

ESCUELA POLITÉCNICA
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA
GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA.



Universidad
Carlos III de Madrid

PROYECTO FIN DE GRADO
ENTORNO DE SIMULACIÓN DE UNITY PRO.
EJEMPLO: SISTEMA DE EMBOTELLADO.

TUTORA: MARÍA DOLORES BLANCO ROJAS.

Álvaro Martínez Galindo.
Junio 2013, Madrid.

RESUMEN.

En los últimos años han ido apareciendo numerosos simuladores en todas las áreas posibles que podamos imaginar: robótica, aviónica, musicales... Y ahora también en la automática, en los PLC.

Schneider Electric ha desarrollado un programa, *Unity Pro*, en el que incluye uno de los primeros simuladores que reproduce el funcionamiento de un autómata. Este trabajo se centra en analizar las opciones y posibilidades que ofrece el programa desde un punto de vista didáctico, dado que se pretende crear tanto una pequeña guía práctica para los alumnos de la asignatura *Automatización Industrial I*, como un programa con el que los alumnos puedan interactuar y ejercitar los conceptos impartidos en la asignatura citada.

Para el ejemplo práctico se ha escogido una planta embotelladora, un proceso sencillo que el alumno no tardará en asimilar, pero con complicaciones para aquellos que quieran dar “un paso más” en la asignatura. Se propone un enunciado, con una solución y una pantalla de operador del diseño. Dicha pantalla ha sido creada gracias a otra de las herramientas que ofrece *Unity Pro*, en ella se podrá ver gráficamente el proceso si se opera con el PLC o bien interactuar con el sistema si se está en modo simulación. La pantalla estará descrita en detalle, a modo de guía para futuras aplicaciones similares.

Habrá también un modo denominado “modo docente”, en él desaparece la solución al problema propuesto y se crea una configuración para llevar el control de la pantalla de operador. En este modo el alumno podrá implementar su solución y ver cómo el sistema realiza el proceso en función de lo que ha programado.

Aparte de diseñar un problema y proponer una solución, con una pantalla de operador sencilla para el alumno, se ha hecho un análisis del programa en sí. Viendo a sus predecesores, el tipo de lenguajes que acepta, variables... y un análisis con las diferencias existentes entre el novedoso simulador y el tradicional autómata.

En resumen, se ha propuesto un problema, programando una solución óptima y un modo docente, destinado a ser un banco de pruebas para los alumnos; junto a todo ello se ha añadido una pantalla de operador que represente totalmente la instalación. Además se ha probado el desarrollo y su evolución gráfica, tanto en el modo simulación como con el PLC, para apreciar las diferencias existentes.

ABSTRACT.

In the last few years, many simulators have been coming into view in a wide variety of fields, such as robotics, avionics, musical simulators, etc...and now it's the turn of automatics.

Schneider Electric has developed a program, Unity Pro, which includes one of the first simulators that faithfully reproduces the operation of an automaton. This work is focused on analyzing the chances of this program from a didactic point of view, as it's expected to provide with, on the one hand, a small practical reference book for the students of the subject called *Industrial Automation I*, and on the other hand, a program where the main concepts of the abovementioned subject can be practiced by the students.

For the practical example, it's been selected a bottling plant. This is an easy process, easily understandable by the students, but including complications for those ones who wanted to delve into the subject. An issue has been proposed, with a solution and an *Operator Screen*. This screen, which has been built by another of the tools offered by Unity Pro, allows monitoring the process if we are working on the PLC, or even interact with the system whenever we are in simulation mode. The Operator Screen will be explained thoroughly, so as it can be used as a reference book for future applications.

There will be a *Teach Mode*, where the solution for the proposed problem disappears and a new configuration in order to take control of the operator screen is established. In this mode, the student will be able to implement his solution and see how the system carries out the process according to the programmed code.

Apart from designing a problem and proposing a solution with an easy Operator Screen, it has been made an analysis of the program Unity Pro. It has been described its predecessors, the type of languages supported, variables... and a brief comparison between the new simulator and the classic PLC.

To sum up, a problem has been proposed, programming the optimal solution and a teach mode aimed at students' testing. Moreover, an *Operator Screen* has been added, so as the facilities can be utterly depicted. Finally, everything has been checked on the PLC and on the simulator, in order to assess the differences among them.

AGRADECIMIENTOS.

Llegados a este punto toca pararse, echar la vista atrás y dar las gracias a todas aquellas personas que han contribuido de una manera u otra a que haya llegado hasta aquí.

En primer lugar, dar las gracias a Dolores, mi tutora, por darme esta oportunidad de realizar el Proyecto Fin de Grado, en el área que más inquietudes ha despertado en mí durante la carrera. Gracias por guiarme en estos últimos pasos, proporcionándome siempre alternativas. Para mí ha sido un placer realizar este proyecto bajo tu supervisión.

La vida te va poniendo personas que entran en ella para quedarse y adquirir una gran importancia. Gracias a Sofía, lo más parecido a una hermana, por aguantarme yendo juntos desde primaria, hasta la misma carrera en la universidad; práctica tras práctica, trabajo tras trabajo, siempre serás “mi tata”. Gracias Borja por todas esas horas en clase llenas de buenos momentos y por estar ahí siempre que se le pedía un favor, un amigo del que por mucho bueno que se diga sería quedarse corto. A David, porque también hemos compartido numerosos momentos juntos y sé que después de estos años me llevo otro amigo que va a estar en las buenas y en las malas, eres una grandísima persona.

Mencionar también a mis amigos del colegio: Pedro, Jorge, Ariane, Alberto, Cristina, Roberto y Rodri; que a pesar de llevar tantos años juntos nuestra amistad sigue como el primer día, vosotros también merecéis una mención.

No me quiero olvidar de “mis chicas de Pinto”, Beita y Mar, mi abogada y mi profe favorita. Son únicos los momentos juntos vividos en Gandía, verano tras verano, y aunque nos vemos con menos frecuencia de la que me gustaría, sé que siempre estáis ahí.

Gracias a mi familia y en especial a mis padres y a “mi Yayo”. Los primeros por enseñarme a esforzarme desde bien pequeño y a no conformarme, inculcándome que con trabajo se llega a donde se desee y ese es el único secreto del éxito. Al segundo por preocuparse por mí preguntándome día tras día, desde que volvía del colegio hasta ahora, cómo me había ido y diciéndome “*tu hijo hazme caso y estudia que todo eso es luego para tí*”.

Por último, y más importante, muchas gracias Mónica por apoyarme en todo, comprenderme y entender que todo el tiempo que muchas veces no he pasado contigo ha dado sus frutos para poder llegar hasta aquí. Gracias de corazón por sacarme una sonrisa en los momentos difíciles, estar día a día conmigo y saber entender a una cabecita tan complicada como la mía, gracias cielo.

*“Una habilidad mediana, con esfuerzo,
llega más lejos en cualquier arte que un talento sin él”
Baltasar Gracián.*

ÍNDICE GENERAL:

RESUMEN.	2
ABSTRACT.	3
AGRADECIMIENTOS.	4
ÍNDICE DE TABLAS.	7
ÍNDICE DE FIGURAS.	8
CAPÍTULO I –MOTIVACIÓN Y OBJETIVOS.	10
1.1 MOTIVACIÓN.	11
1.2 OBJETIVOS.	11
1.3 ESTRUCTURA DEL DOCUMENTO.	12
CAPÍTULO II –UNITY PRO.	14
2.1 HISTORIA.	15
2.1.1 Autómata programable.	15
2.1.2 Schneider Electric.	16
2.2 PRECEDENTES.	18
2.3 PROGRAMA UNITY PRO.	20
2.3.1 Tipos de variables, direcciones, bits y palabras del sistema.	24
2.3.1.1 Variables.	24
2.3.1.2 Direcciones.	26
2.3.1.3 Bits y palabras de sistema.	26
2.3.2 Simulador del Autómata.	28
2.3.3 Editor de ventanas de usuario.	29
CAPÍTULO III –EJEMPLO DESARROLLADO.	31
3.1 ENUNCIADO DEL PROBLEMA PROPUESTO.	32
3.2 VARIABLES DEL PROCESO.	36
3.2.1 Variables de entrada y salida.	36
3.2.3 Variables generadas por el programa.	37
3.2.3 Otras variables.	38
3.3 SECCIONES DEL PROGRAMA.	39
3.3.1 Embotelladora.	39
3.3.2 Control del depósito.	41
3.3.3 Parada de emergencia.	41
3.3.4 Reinicio de contadores.	42
3.4 PROGRAMACIÓN.	43
CAPÍTULO IV –PANTALLA DE OPERADOR.	47
4.1 DEFINICIÓN Y CREACIÓN.	48
4.2 EXPLICACIÓN DE ELEMENTOS.	50
4.3 EJEMPLOS DE ELEMENTOS CREADOS.	53
4.3.1 Cuadro de mandos.	53
4.3.2 Simulación de sensores.	55
4.3.3 Simulación de salidas.	56
4.3.4 Simulación del nivel del depósito.	57

4.3.5 Simulación del movimiento de botellas.	58
CAPÍTULO V –EVOLUCIÓN DEL SISTEMA.	59
CAPÍTULO VI –MODO DOCENTE.	64
CAPÍTULO VII –COMPARATIVA MODO SIMULACIÓN - PLC.	67
7.1 DESCRIPCIÓN DEL PLC USADO.	68
7.2 TRANSFERENCIA DEL PROYECTO.	69
7.3 COMPARATIVA SIMULADOR – PLC.	70
7.4 DIFERENCIAS ENTRE EL SIMULADOR Y UN SISTEMA SCADA.	71
CAPÍTULO VIII –PRESUPUESTO.	72
CAPÍTULO IX –CONCLUSIONES Y APLICACIONES FUTURAS.	75
9.1 CONCLUSIONES.	76
9.2 APLICACIONES FUTURAS.	77
10. BIBLIOGRAFÍA.	78
ANEXO I - PROGRAMACIÓN MODO DOCENTE.	79
ANEXO II - DOCUMENTACIÓN TÉCNICA DEL PROYECTO.	84

ÍNDICE DE TABLAS.

TABLA 1. TIPOS DE VARIABLES.	25
TABLA 2. TIPOS DE DIRECCIONES.	26
TABLA 3. VARIABLES DE ENTRADA.	36
TABLA 4. VARIABLES DE SALIDA.	36
TABLA 5. VARIABLES DE TRANSICIONES.	37
TABLA 6. VARIABLES DE ESTADO.....	37
TABLA 7. OTRAS VARIABLES.	38
TABLA 8. EVOLUCIÓN DE LA SIMULACIÓN.	63
TABLA 9. RESUMEN PRESUPUESTO.	74

ÍNDICE DE FIGURAS.

FIGURA 1. UNITY PRO Y PREDECESORES.	18
FIGURA 2. VENTANA DEL PROGRAMA MODSOFT.	18
FIGURA 3. LOGOTIPO DEL SOFTWARE CONCEPT.	19
FIGURA 4. LOGOTIPO DEL SOFTWARE PL7.	19
FIGURA 5. EJEMPLO DE PROGRAMACIÓN EN ST.	20
FIGURA 6. EJEMPLO DE PROGRAMACIÓN EN IL.	21
FIGURA 7. EJEMPLO DE PROGRAMACIÓN EN LD.	21
FIGURA 8. EJEMPLO DE PROGRAMACIÓN EN SFC.	22
FIGURA 9. EJEMPLO DE PROGRAMACIÓN EN FBD.	22
FIGURA 10. SOFTWARE UNITY PRO.	23
FIGURA 11. VENTANA INSPECCIÓN DE ELEMENTOS.	23
FIGURA 12. SIMULACIÓN VIRTUAL DEL PLC.	28
FIGURA 13. EJEMPLO DE VENTANA DE USUARIO.	29
FIGURA 14. ESQUEMA GRÁFICO DE LA INSTALACIÓN.	32
FIGURA 15. DIAGRAMA SFC EMBOTELLADORA.	40
FIGURA 16. DIAGRAMA SFC CONTROL DEPÓSITO.	41
FIGURA 17. DIAGRAMA PARADA DE EMERGENCIA.	42
FIGURA 18. DIAGRAMA RESET DE CONTADORES.	42
FIGURA 19. PROGRAMACIÓN POSICIÓN BOTELLAS.	43
FIGURA 20. PROGRAMACIÓN RESTRICCIONES DE POSICIÓN.	43
FIGURA 21. PROGRAMACIÓN REINICIO DE CONTADORES.	44
FIGURA 22. CONVERSIÓN DE TIME A INT.	44
FIGURA 23. PROGRAMACIÓN CUENTA ATRÁS ETIQUETADORA.	44
FIGURA 24. CONFIGURACIÓN DE CONDICIÓN EN SFC.	45
FIGURA 25. CONTAJE DE TIEMPO DE PARADA.	46
FIGURA 26. PROGRAMACIÓN TIEMPO AUXILIAR.	46
FIGURA 27. CREACIÓN DE UNA PANTALLA DE OPERADOR.	48
FIGURA 28. GRÁFICO DE LA INSTALACIÓN.	49
FIGURA 29. LIBRERÍA DE PANTALLA DE OPERADOR.	50
FIGURA 30. BARRA DE HERRAMIENTAS DE LA PANTALLA DE OPERADOR.	51
FIGURA 31. GRÁFICO CUADRO DE MANDOS.	53

FIGURA 32. CONFIGURACIÓN DE BOTÓN DE COMANDO.	53
FIGURA 33. ANIMACIÓN DE BOTONES.	54
FIGURA 34. CONFIGURACIÓN DE UN CAMPO DE ENTRADA.	54
FIGURA 35. ELEMENTO AGRUPADO Y DESAGRUPADO.	55
FIGURA 36. CREACIÓN DE LA TAPONADORA.	56
FIGURA 37. CONFIGURACIÓN DE UN GRÁFICO DE BARRAS.	57
FIGURA 38. CONFIGURACIÓN SIMULACIÓN DE BOTELLAS.	58
FIGURA 39. EVOLUCIÓN GRÁFICA DE LA BOTELLA.	58
FIGURA 40. ASPECTO DE LA INSTALACIÓN EN REPOSO.	60
FIGURA 41. INICIO DEL PROCESO.	61
FIGURA 42. BOTELLA LLENA.	61
FIGURA 43. PROCESO DE TAPONADO.	61
FIGURA 44. SECUENCIA DE ETIQUETADO.	61
FIGURA 46. OPCIONES CONTROL CALIDAD.	62
FIGURA 47. BOTELLA EN ZONA DE EMPAQUETADO.	62
FIGURA 48. ACTIVACIÓN DE LA EMPAQUETADORA.	62
FIGURA 49. BOTELLA EN ZONA DE DESECHO.	62
FIGURA 50. LÍMITE DE BOTELLAS DESECHADAS.	62
FIGURA 51. VACIADO DEL DEPÓSITO.	63
FIGURA 52. ELEMENTO DE ALARMA.	63
FIGURA 53. CÓDIGO SECCIÓN CONDICIÓN, MODO DOCENTE.	66
FIGURA 54. CONTROL DEL DISPLAY ETIQUETADORA, MODO DOCENTE.	66
FIGURA 55. AUTÓMATA DEL LABORATORIO DE AUTOMATIZACIÓN DE UC3M.	68
FIGURA 56. PESTAÑA DE GENERAR PROYECTO.	69
FIGURA 57. CONFIGURACIÓN DE DIRECCIÓN.	69
FIGURA 58. PROCESO PARA TRANSFERIR UN PROYECTO.	70

CAPÍTULO I – Motivación y objetivos.

1.1 Motivación.

Dentro de la continua renovación y con objeto de ofrecer a los alumnos las últimas y mejores herramientas al alcance de la universidad, el departamento de Ingeniería de Sistema y Automática decidió sustituir el software existente en los laboratorios de automatización, utilizados en la asignatura de *Automatización Industrial I*. En dichos laboratorios encontramos lo autómatas *TSX Premium de Schneider* que en años anteriores eran programados por los alumnos utilizando el software *PL7 Junior*, pero que este mismo año ha sido reemplazado por el programa *Unity Pro*, desarrollado por la misma empresa *Schneider Electric*.

La nueva herramienta ofrece todas las posibilidades que ofrecía su antecesor y otras nuevas que se explicarán con más detalle en este documento, la más novedosa es que los alumnos dispondrán de un simulador presente en propio programa, lo que hace que se puedan llevar a cabo pruebas sobre la programación en cualquier ordenador, sin necesidad de estar conectado al autómata. De esta forma se eleva exponencialmente el número de puestos para que los alumnos puedan probar sus diseños y aplicar sus conocimientos. Antes se limitaba a las aulas del laboratorio de automatización, con sus respectivos horarios de apertura, y ahora disponen del software en cualquier ordenador de la universidad, por lo que podrán probarlos antes de llevarlos al autómata; abriendo un gran abanico de posibilidades para la asignatura gracias a este cambio.

En este abanico se plantea dar a los alumnos la posibilidad de interactuar con el programa, no sólo de la manera habitual, programar el SFC o su hoja de lenguaje de contactos y comprobar el resultado, se busca una nueva forma más dinámica y con mayor potencial didáctico: crear una interfaz gráfica que responda a lo programado, de tal forma que puedan ver en ella la ejecución del programa que estén desarrollando y puedan detectar y analizar todos los fallos. Con todo ello se pretende que alumno se involucre más y aumente su motivación por la asignatura.

1.2 Objetivos.

Este trabajo se centra en las posibilidades didácticas que ofrece este nuevo software y se ha enfocado con tres objetivos principales:

- Proponer un problema junto con una solución eficaz.
Se redactará un enunciado para que los alumnos lo resuelvan, basado en un sistema de embotellado genérico; además se dará una solución eficaz que comprenda todos los conocimientos adquiridos en diversas asignaturas del Grado, junto con otros que han derivado al profundizar más en la materia al realizar este proyecto.
- Creación de una pantalla de operador.
Junto con el problema propuesto, se ha aprovechado la posibilidad que ofrece *UnityPro* de crear una pantalla de operador (interfaz gráfica) donde ver e interactuar con el proceso.

- Implementación de un “Modo Docente”.
Se tratará de aislar el comportamiento de la pantalla de operador de la solución del problema. De esta forma los alumnos podrán introducir su diseño y ver si el desarrollo es el esperado o no.
- Comparar resultados con el simulador y con el Autómata.
Dado que no se ha probado hasta ahora, implementar la programación en modo simulación y en modo autómata y ver las posibles diferencias.

1.3 Estructura del documento.

Aparte del presente capítulo, en el que se expone la motivación y los objetivos perseguidos, el contenido del Proyecto Fin de Grado se divide en otros ocho capítulos, más la bibliografía empleada.

- El siguiente capítulo, numerado como Capítulo 2 - Unity Pro, presenta el software utilizado en la realización del presente trabajo: *Unity Pro*. Se describe la historia de los PLC's y de la empresa Schneider Electric (creadora del programa). Se analiza las características del software: tipo de lenguaje que admite, variables... Además de describir brevemente el simulador y la herramienta de pantallas de operador.
- En el Capítulo 3 - Ejemplo desarrollado, se describe el problema planteado: un sistema de embotellado. Además se presenta un enunciado de dicho sistema, los diagramas SFC, las variables empleadas y los aspectos más complejos de la programación que se ha llevado a cabo.
- En el Capítulo 4 – Pantalla de operador, se define cómo crear una pantalla de operador y se muestra la creada para el ejemplo de sistema de embotellado. Se presentan las opciones que presenta esta herramienta y se describe paso por paso cómo se han creado los elementos de la pantalla diseñada.
- En el Capítulo 5 – Evolución del sistema, en él se detalla paso por paso el aspecto que va tomando la simulación y el orden de los elementos que se han de ir accionando. Mostrándose el estado de la pantalla de operador en cada momento y en función de las diferentes opciones posibles.
- En el Capítulo 6 – Modo Docente, se explica que este modo es un modo específico para que el alumno pueda programar sobre él e ir viendo en la pantalla de operador si el resultado es satisfactorio. Se dan aclaraciones de los pasos dados y las peculiaridades del diseño de este modo.

- En el Capítulo 7 – Comparativa modo simulación – PLC, lo primero que se hace en este apartado es describir el autómatas presente en los laboratorios de Automatización. Posteriormente, se elabora un análisis de las diferencias existentes entre ejecutar un proceso en el simulador y las diferencias respecto a ese mismo proceso procesado por un PLC tradicional. Para terminar se abordan los elementos diferenciadores entre el simulador y un sistema SCADA.
- En el Capítulo 8 – Presupuesto, se elabora un breve presupuesto en el que se describe el autómatas y el software junto con la cantidad necesaria para adquirir el autómatas completo y el software *Unity Pro*.
- Por último, en el Capítulo 9 – Conclusiones y aplicaciones futuras, se presentan las conclusiones derivadas del uso y manejo del software. Además se tratan las aplicaciones futuras: por un lado posibles ampliaciones del sistema de embotellado propuesto; por otro, las aplicaciones docentes y profesionales que puede ofrecer *Unity Pro*.
- Aparte de los capítulos explicados tenemos dos anexos. El primero de ellos, Anexo I – Programación Modo Docente incluye el código generado para poder controlar el simulador y la pantalla de operador de manera independiente para poder operar según el denominado *modo docente*. En el Anexo II – Documentación Técnica del proyecto, se adjunta toda la documentación generada por el programa *Unity Pro* para el proyecto realizado.

CAPÍTULO II – Unity Pro.

2.1 Historia.

2.1.1 Autómata programable.

Los primeros PLC's (Controlador lógico programable - *Programmable Logic Controller*) en la industria datan de la década de 1960. La razón principal fue la necesidad de eliminar el alto coste que se producía al reemplazar el sistema de control basado en relés y contactos. La solución la aportó Bedford Associates, que propuso algo denominado Controlador Digital Modular (*MODICON, MODular Digital CONtroller*) a un gran fabricante de coches. Otras compañías propusieron esquemas basados en ordenador, uno de los cuales estaba basado en el PDP-8. El MODICON 084 resultó ser el primer PLC del mundo en ser producido comercialmente. El problema de los relés era que cuando los requerimientos de producción cambiaban también lo hacía el sistema de control. Esto comenzó a resultar bastante caro cuando los cambios fueron frecuentes. Dado que los relés son dispositivos mecánicos y poseen una vida limitada se requería un mantenimiento planificado. Por otra parte, a veces se debían realizar conexiones entre cientos o miles de relés, lo que implicaba un enorme esfuerzo de diseño y mantenimiento. Los nuevos controladores debían ser fácilmente programables por ingenieros de planta o personal de mantenimiento.

El tiempo de vida debía ser largo y los cambios en el programa tenían que realizarse de forma sencilla. Finalmente se imponía que trabajaran sin problemas en entornos industriales adversos. La solución fue el empleo de una técnica de programación familiar y reemplazar los relés mecánicos por relés de estado sólido.

A mediados de los 70 las tecnologías dominantes de los PLC eran máquinas de estados secuenciales y CPU basadas en desplazamiento de bit. Los AMD 2901 y 2903 fueron muy populares. Los microprocesadores convencionales proveyeron la potencia necesaria para resolver de forma rápida y completa la lógica de los pequeños PLC's. Por cada modelo de microprocesador había un modelo de PLC basado en el mismo. No obstante, el 2903 fue de los más utilizados.

La capacidad de comunicación comenzó a aparecer en 1973 aproximadamente. El primer sistema fue el bus Modicon (Modbus). El PLC podía ahora dialogar con otros PLC y ubicarse alejado de las máquinas que controlaba. También se introdujeron en el mundo analógico al poder enviar y recibir señales de tensión. Desafortunadamente, la falta de un estándar acompañado de un continuo cambio tecnológico ha hecho que la comunicación de PLC sea un cúmulo inconexo de sistemas físicos y protocolos incompatibles entre sí.

En los años 80 se produjo un intento de estandarización de las comunicaciones con el MAP (*Manufacturing Automation Protocol*) de General Motor's. También se redujeron las dimensiones del PLC y se pasó a programar con lenguajes simbólicos a través de ordenadores personales en vez de los clásicos terminales de programación. Los años 90 mostraron una gradual reducción en el número de nuevos protocolos de comunicación, y en la modernización de las capas físicas de los protocolos más populares que sobrevivieron a los 80. El estándar IEC 1131-3 intenta unificar el sistema de programación de todos los PLC en un único estándar internacional. Ahora disponemos de PLC's que pueden ser programados en diagramas de bloques, lista de instrucciones, C y texto estructurado al mismo tiempo.

Los PC están comenzando a reemplazar al PLC en algunas aplicaciones, incluso la compañía que introdujo el Modicon 084 ha cambiado al control basado en PC. [8]

2.1.2 Schneider Electric.

Este software esta desarrollado por la compañía Schneider Electric, tiene sus orígenes en 1836 en la compra por parte de Adolphe y Eugène Schneider de las minas e instalaciones siderúrgicas de Le Creusot. En esos momentos el eje de las actuaciones empresariales era la siderurgia, adquiriendo una creciente presencia internacional, especialmente a partir de 1870, teniendo una de las principales vías de especialización en la producción de armas pesadas. [1]

“Schneider”, después de la Segunda Guerra Mundial, vivió un intenso proceso de reformulación de su proyecto empresarial, dirigiéndose cada vez más al mercado civil. El cambio no fue del todo exitoso en un contexto de crisis de la siderurgia y de la construcción naval provocando cambios en la propiedad de la ya entonces centenaria firma. Esta situación se concretó con la crisis de Creusot-Loire. Un nuevo giro en la empresa se produjo a inicios de la década de 1980. A partir de 1986 existía el convencimiento de que no se podía dejar sola a “Merlin Gerin”. En ese marco se inició una simplificación de las estructuras financieras del grupo. En esos momentos Schneider S.A. tenía tres puntos de apoyo: “Spie-Batignolles”, “Merlin Gerin” y “Jeumont- Schneider”. Los tres estaban directamente relacionados con la producción y/o distribución de energía eléctrica, aunque con importantes diferencias entre ellos. Esa sería la base para la reorganización empresarial de “Schneider”. En ese contexto se produjo la adquisición de “Telemecanique” en 1988 y “Square D” en 1991. Ello formaba parte de la estrategia de centrar la actuación en el sector eléctrico. El cambio o el despliegue se tradujo en la nueva orientación hacia el mercado eléctrico y la entrada en competencia internacional con los grandes grupos. Los primeros tiempos de convivencia entre “Merlin Gerin” y “Telemecanique” no fueron nada fáciles. Con todo, con las operaciones realizadas entres 1987 y 1991 el “Grupo Schneider” dobló su tamaño. En 1993 ya ocupaba el número 16 en el *ranking* de empresas francesas y el 52 a escala europea. En ese contexto se produjo la necesidad de reorganizar internamente al grupo y simplificar su estructura. La pieza clave de este proyecto, datado en 1992 fue la fusión entre “Schneider S.A.” y “Merlin Gerin S.A.” Se constituyeron cuatro áreas de gestión: “Schneider North America”, “Schneider Europe” y “Schneider International”, junto a la dedicada a la gestión en Francia. Fruto de esa operación HIMEL pasó a estar identificada como miembro del “Grupo Schneider”. A partir de mayo de 1999 el “Groupe Schneider” adoptó el nombre de “Schneider Electric” con ello finaliza el período de refocalización en el negocio eléctrico.

“Schneider Electric” después de esta evolución se ha convertido en una empresa líder en el “Power & Control”. Sus ámbitos de actuación son fundamentalmente dos: distribución eléctrica y automatismos/control. Recientemente se ha creado una nueva unidad de negocio “Critical Power & Cooling”. Sus actividades se agrupan en cinco grandes mercados: el de media tensión, el de baja tensión, el control industrial, los automatismos programables y los servicios. Es, sin duda alguna, un grupo líder mundial en esos mercados. Las ventas del grupo entre 2003 y 2007 se han multiplicado por dos. “Schneider Electric” está presente, con producción o comercialización, en 130 países y cuenta con una plantilla en torno de las 120.000 personas.

Desde el punto de vista de organización el grupo hasta hace bien poco se estructuraba en cuatro divisiones: Asia-Pacífico con sede en Hong Kong, Europa (sin la Península Ibérica) en París, América del Norte en Chicago e “Internacional & Ibérica” con administración desde Barcelona. Esta última división tenía competencia sobre la Península Ibérica, Oriente Medio, África, América del Sur y el Caribe. Ello supone presencia en más de 100 países, 26 centros industriales y 27 centros logísticos. A nivel estratégico “Schneider Electric” está centrando su interés inversor en los países emergentes, especialmente en la China donde el grupo tiene más de veinte factorías.

Esta empresa también tiene una notable participación en nuestro país. Sus orígenes se remontan a principios del S.XX, por aquel entonces la compañía creada en Barcelona, con capital suizo, se denominaba “Sociedad Española Gardy S.A”, dedicada a la fabricación de material eléctrico; que posteriormente y tras diversos cambios, pasa a ser “Schneider Electric España S.A.” con la mayor parte del capital en manos francesas y suizas. Desde 2008 en Barcelona se sitúa la sede del “European Operacional Division” (EOD). Ello supone que desde la ciudad catalana, se administra alrededor de un tercio del negocio total del grupo francés. A finales del 2010 “Schneider Electric España” tiene una plantilla cercana a los 2.500 efectivos distribuidos y unos ingresos de explotación de 839,65 millones de euros. Su actividad se distribuye entre 10 centros productivos (Capellades, Molins de Rei (Barcelona); Burlada, Puente la Reina (Navarra); Griñon (Madrid); Munguía, Gatica (Vizcaya); Valencia, Corbera, Meliana (Valencia). Además cuenta con el centro logístico en Sant Boi de Llobregat (Barcelona), anteriormente nombrado. ^[2]

2.2 Precedentes.

Unity Pro surge de la integración del software **Concept**, con el que se podía programar los PLC's Quantum y Momentum, y del software **PL7 Pro**, con el que se podía programar los PLC's Micro y Premium. Anteriormente a Concept existía un software llamado **Modsoft**.

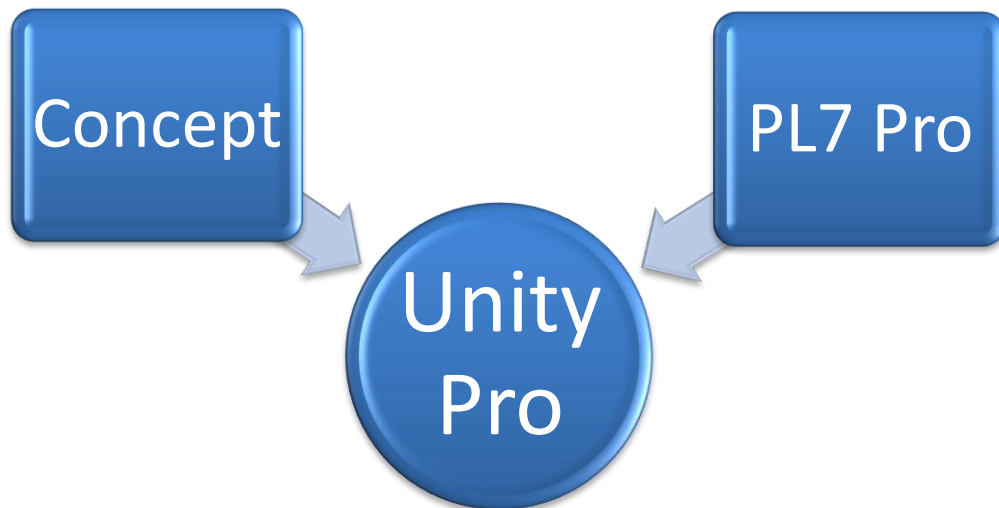


Figura 1. Unity Pro y predecesores.

Cronológicamente hablando, el primer paso para llegar hasta el actual *Unity Pro* fue el software **Modsoft**. El cual es un entorno de programación integrado diseñado como aplicación MS-DOS. Permite crear configuraciones de lógica de contactos 984 y aplicaciones para los PLC's Modicom 984. Si se utiliza la última versión, todos los programas creados en Modsoft para 984 pueden ejecutarse en las CPU Quantum. Admite dos modos de programación: en línea y sin conexión; además ofrece numerosas posibilidades de documentación (listados de red con referencias cruzadas, configuración del PLC, archivos de comentarios...).^[3]

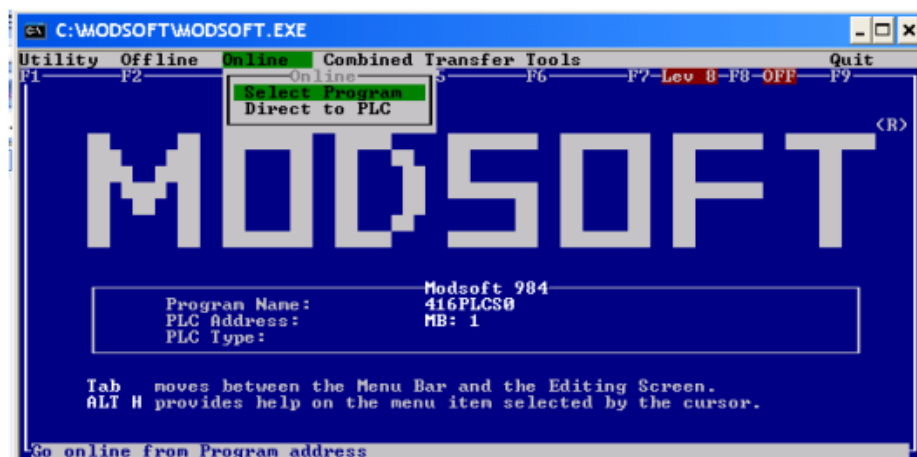


Figura 2. Ventana del programa MODSOFT.

El sucesor de *Modsoft* fue **Concept**, el principio que guía su desarrollo es que todo el sistema de procedimientos de configuración y todos los editores deben tener la misma apariencia. La mayoría de los pasos de configuración, en especial la creación del programa, se han diseñado de forma independiente del PLC a programar. Concept es la herramienta de proyección unificada para Quantum, Compact, Momentum, Atrium y productos SoftPLC. En lo referente a los lenguajes de programación admite los siguientes: FBD (Function Block Language), LD (Ladder Diagram), SFC (*Sequence language*), IL (*Instruction List*), ST (Structured Text) y LL984 edited (*Modsoft orientated Ladder Logic*); pudiéndose combinar entre ellos. Permite, como su predecesor, programación tanto en línea (*online*) como desconectado (*offline*).^[4]

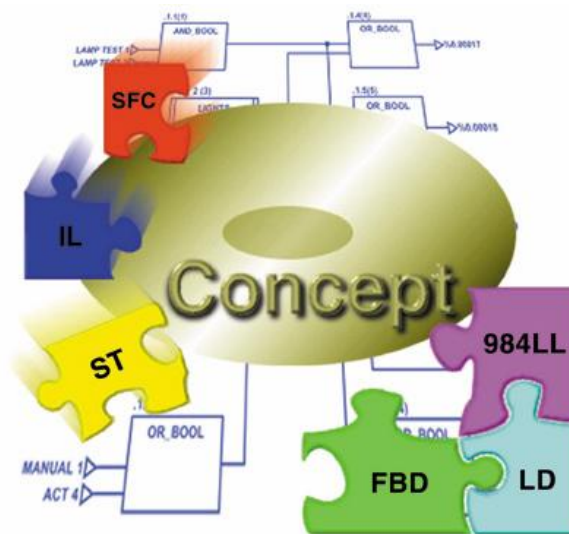


Figura 3. Logotipo del software CONCEPT.

El principal programa del que deriva *Unity Pro* es la serie denominada **PL7** con sus respectivas versiones (Junior, Profesional...); este era el software presente en los laboratorios de automática en su versión Junior. Al igual que *Concept*, se puede programar en cinco lenguajes diferentes: LD, ST, IL, SFC y FBC; contando con una amplia biblioteca de funciones y de bloques ya incluidos en el propio programa. Presenta funciones sofisticadas como rearranque en caliente y en frío, posibilidad de realizar multitareas, tareas maestras, eventos... Además de un modo de depuración.



Figura 4. Logotipo del software PL7.

2.3 Programa Unity Pro.

El software de programación Unity Pro XL es un software de última generación y de reciente implantación en el campo de la automatización industrial.

El software tiene como objetivo la programación y configuración de autómatas. Con esta herramienta se configura el hardware que forma el PLC y la programación del procesador para que ejecute las instrucciones en función del proyecto. Dentro del software Unity Pro XL podemos generar proyectos en varias familias de autómatas, como son Modicom M340, Premium, Quantum y Atrium. Todos ellos dentro de la marca Schneider Electric. Cumpliendo la **norma IEC 61131-3** referente a los lenguajes de programación, el propósito de dicha norma es estandarizar la programación de PLC's según el proceso y no el fabricante. Esto quiere decir, que el código creado según esta norma en el software Unity Pro XL de Schneider Electric es totalmente exportable al software de cualquier otro fabricante, el cual también cumpla esta norma como puede ser el caso de Siemens con su software de programación STEP – 7. El uso de dicha norma proporciona una serie de ventajas entre las cuales se encuentran:

- Se reduce el gasto en recursos humanos, formación, mantenimiento y consultoría.
- Evita las fuentes habituales de problemas por el alto nivel de flexibilidad y reusabilidad del software.
- Las técnicas de programación son utilizables en amplios sectores (control industrial en general).
- Combinan adecuadamente diferentes elementos que pueden provenir de diferentes fabricantes, programas, proyectos...
- Incrementa la conectividad y comunicación entre los distintos departamentos y compañías.

Este software permite la programación de los autómatas en los siguientes lenguajes, todos ellos estandarizados mediante la norma anteriormente nombrada: ^[5]

- Literales:

- **ST. - Texto Estructurado (Structured Text).**

Es un lenguaje de alto nivel con orígenes en ADA, Pascal, y 'C'; puede ser utilizado para codificar expresiones complejas e instrucciones anidadas; este lenguaje dispone de estructuras para bucles (*REPEAT-UNTIL*; *WHILE-DO*), ejecución condicional (*IF-THEN-ELSE*; *CASE*), funciones (*SQRT*, *SIN*, etc.).

```
IF ( BotonPartir OR Motor ) AND NOT BotonParar THEN
    Motor := TRUE;
ELSE
    Motor := FALSE;
END_IF;
```

Figura 5. Ejemplo de programación en ST.

➤ **IL.- Lista de Instrucciones (Instruction List).**

Es el modelo de lenguaje ensamblador basado en un acumulador simple; procede del alemán *Anweisungsliste*, *AWL*.

En los autómatas de gama baja, es el único modo de programación. Consiste en elaborar una lista de instrucciones o nemónicos que se asocian a los símbolos y su combinación en un circuito eléctrico a contactos. También decir, que este tipo de lenguaje es, en algunos los casos, la forma más rápida de programación e incluso la más potente.

000	LD	%I0.1	Bp. inicio ciclo
	AND	%I0.0	Dp. presencia vehículo
	AND	%M3	Bit autorización reloj calendario
	AND	%I0.5	Fc. alto rodillo
	AND	%I0.4	Fc. detrás pórtico
005	S	%M0	Memo inicio ciclo
	LD	%M2	
	AND	%I0.5	
	OR	%I0.2	Bp. parada ciclo
	R	%M0	
010	LD	%M0	
	ST	%Q0.0	Piloto ciclo

Figura 6. Ejemplo de programación en IL.

- Gráficos.

- **LD.- Diagrama de Contacto (Ladder).**

Es el que más similitudes tiene con el utilizado por un electricista al elaborar cuadros de automatismos. Muchos autómatas incluyen módulos especiales de software para poder programar gráficamente de esta forma.

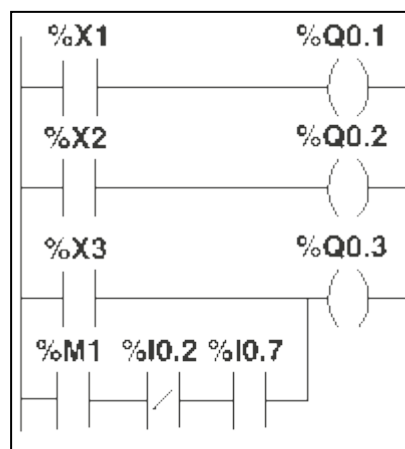


Figura 7. Ejemplo de programación en LD.

- **SFC. - Grafcet** (Secuencial Funtion Chart).

Es el llamado Gráfico de Orden Etapa Transición. Ha sido especialmente diseñado para resolver problemas de automatismos secuenciales. Las acciones son asociadas a las etapas y las condiciones a cumplir a las transiciones. Este lenguaje resulta enormemente sencillo de interpretar por operarios sin conocimientos de automatismos eléctricos. Muchos de los autómatas que existen en el mercado permiten la programación en GRAFCET, tanto en modo gráfico o como por lista de instrucciones. También podemos utilizarlo para resolver problemas de automatización de forma teórica y posteriormente convertirlo a plano de contactos.

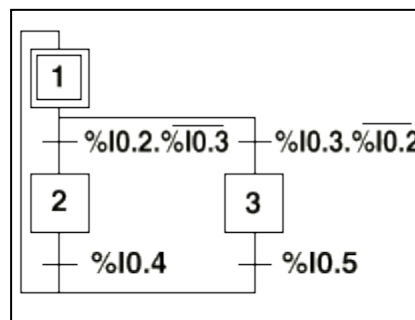


Figura 8. Ejemplo de programación en SFC.

- **FBD. - Lenguaje de bloques funcionales** (Funtional Block Diagram).

El plano de funciones lógicas, resulta especialmente cómodo de utilizar, a técnicos habituados a trabajar con circuitos de puertas lógicas, ya que la simbología usada en ambos es equivalente.

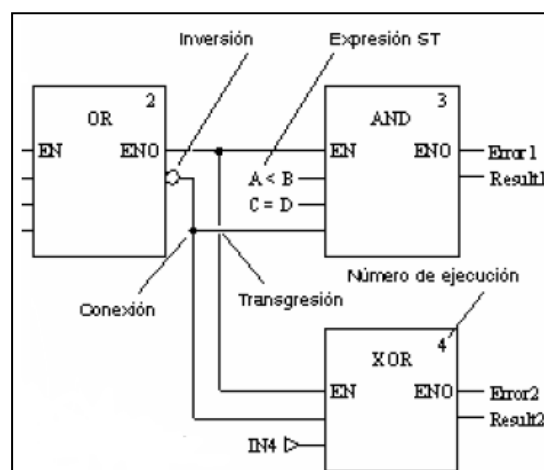


Figura 9. Ejemplo de programación en FBD.

Aparte de poderse programar en estos cinco lenguajes explicados, este software presenta numerosas características, algunas de las que destacan son las siguientes:

- Está basado en formatos estándares, reduciendo sensiblemente las actividades de familiarización y capacitación.
- Presenta las herramientas necesarias para la creación, depuración y puesta en marcha de las aplicaciones.
- El entorno de ejecución de los programas es Windows 98/2000/NT/XP/7, estando adaptado a su funcionamiento gráfico y orientado a objetos.
- Su utilización es intuitiva y por lo tanto fácil.
- Ventana de diagnóstico integrado.



Figura 10. Software Unity Pro.

La principal ventaja de este software de programación es la estandarización de sus códigos gracias al cumplimiento de la norma anteriormente citada IEC 61131-3, con la cual se consigue que *Unity Pro* proponga un conjunto completo de funcionalidades y de herramientas que permiten calcar la estructura de la aplicación en la estructura del proceso o de la máquina. El programa se divide en módulos funcionales jerarquizados que agrupan:

- Secciones de programa.
- Tablas de animación.
- Pantallas de los operadores.
- Hipervínculos.

Unity Pro ofrece además una biblioteca de DFB (Bloques de Funciones de Usuario) ya programados como son las funciones de diagnóstico de aplicaciones, los cuales ayudan a reducir los tiempos de paradas de las instalaciones. Dichos bloques se encuentran ya integrados en el programa y, según su función, permiten vigilar las condiciones permanentes de seguridad y la evolución del proceso en el tiempo. Una ventana de visualización muestra, de forma clara y cronológicamente, con marcación de tiempo en origen, todos los fallos del sistema y de la aplicación. Desde esta ventana, se accede mediante un simple clic al editor de programa en el que se ha producido el error (búsqueda en el origen de las condiciones que faltan).

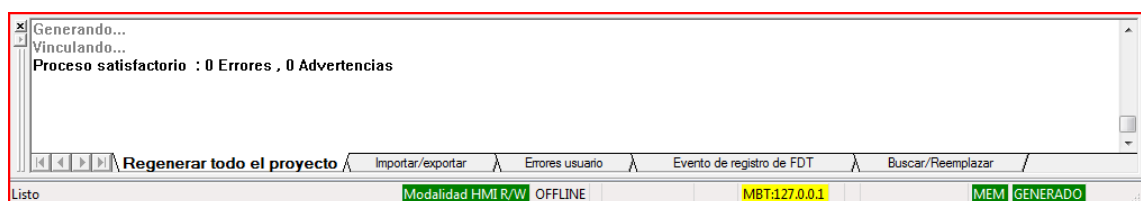


Figura 11. Ventana inspección de elementos

2.3.1 Tipos de variables, direcciones, bits y palabras del sistema.

2.3.1.1 Variables.

Unity Pro acepta diversos tipos de variables, las cuales se pueden definir todas al inicio del programa o bien según se vayan necesitando. Las más importantes son aquellas que funcionen como entrada o salida de nuestro autómata. Las posibilidades, más destacadas, entre las que se puede elegir son las siguientes: ^[10]

TIPO	DESCRIPCIÓN
BOOL	Sigla del tipo booleano. Se trata del elemento de datos de base en informática. Una variable de tipo BOOL posee uno de estos valores: 0 (FALSE) o 1 (TRUE). Un Bit extraído de la palabra es de tipo BOOL, por ejemplo: %MW10.4.
DATE	El tipo DATE codificado en BCD en un formato de 32 bits que contiene esta información: <ul style="list-style-type: none"> • El año codificado en un campo de 16 bits • El mes codificado en un campo de 8 bits • El día codificado en un campo de 8 bits El tipo DATE debe introducirse así: D#<Año>-<Mes>-<Día>
DINT	Sigla del formato Double INTeger (entero doble) (codificado en 32 bits). Los límites inferior y superior figuran a continuación: de -(2 elevado a 31) a (2 elevado a 31) -1. Ejemplo: -2.147.483.648, 2.147.483.647, 16#FFFFFFFF.
DT	Sigla de Date and Time (fecha y hora). El tipo DT, codificado en BCD en un formato de 64 bits, contiene esta información: <ul style="list-style-type: none"> • El año codificado en un campo de 16 bits • El mes codificado en un campo de 8 bits • El día codificado en un campo de 8 bits • La hora codificada en un campo de 8 bits • Los minutos codificados en un campo de 8 bits • Los segundos codificados en un campo de 8 bits Nota: No se utilizan los 8 bits menos significativos. El tipo DT debe
WORD	El tipo WORD se codifica en un formato de 16 bits y se utiliza para realizar tratamientos en las cadenas de bits.

EBOOL	<p>Sigla del tipo Extended BOOLEAN (booleano extendido). Una variable de tipo EBOOL posee el valor 0 (FALSE) o 1 (TRUE), pero igualmente los flancos ascendentes o descendentes y las funciones de forzado.</p> <p>Una variable de tipo EBOOL ocupa un byte de memoria.</p> <p>El byte se compone de los siguientes elementos:</p> <ul style="list-style-type: none"> • Un bit del valor • Un bit del historial (cada vez que cambia el objeto de estado, el valor se copia en el bit del historial) • Un bit del forzado (igual a 0, si el objeto no se fuerza, igual a 1 si el bit se fuerza) • El valor predeterminado de cada bit es 0 (FALSE).
INT	<p>Sigla del formato single INTeger (entero simple) (codificado en 16 bits). Los límites inferior y superior figuran a continuación: de $-(2^{15})$ a $(2^{15}) - 1$.</p>
REAL	<p>El tipo REAL (real) es un tipo codificado en 32 bits. Los rangos de valores posibles se detallan a continuación:</p> <p>Cuando un resultado:</p> <ul style="list-style-type: none"> • Esta comprendido entre $-1,175494e-38$ y $1,175494e-38$, se le considera como DEN; • Es inferior a $-3,402824e+38$, aparece el símbolo -INF (de -infinito); • Es superior a $+3,402824e+38$, aparece el símbolo INF (de +infinito); • Está indefinido (raíz cuadrada de un número negativo), aparece el símbolo NAN
STRING	<p>Una variable de tipo STRING es una cadena de caracteres ASCII. La longitud máxima de una cadena de caracteres es de 65.534 caracteres.</p>
TIME	<p>El tipo TIME expresa la duración en milisegundos. Codificado en 32 bits, este tipo permite obtener duraciones de 0 a $2^{32}-1$ milisegundos. Las unidades de tipo TIME son las siguientes: días (d), horas (h), minutos (m), segundos (s) y milisegundos (ms). Un valor literal de tipo TIME se representa mediante una combinación de tipos anteriores que preceden a T#, t#, TIME# o time#.</p> <p>Ejemplos: T#25h15m, t#14,7S, TIME#5d10h23m45s3ms</p>
TOD	<p>Sigla de Time Of Day (hora del día).</p> <p>El tipo TOD, codificado en BCD en un formato de 32 bits, contiene esta información:</p> <ul style="list-style-type: none"> • La hora codificada en un campo de 8 bits • Los minutos codificados en un campo de 8 bits • Los segundos codificados en un campo de 8 bits <p>Nota: No se utilizan los 8 bits menos significativos. El tipo Time Of Day debe introducirse así: TOD#<Hora>:<Minutos>:<Segundos></p>

TABLA 1. Tipos de variables.

2.3.1.2 Direcciones.

Respecto a las direcciones se debe conocer cuáles son los tipos y características de cada unas de ellas para poder implementar mejor el desarrollo del programa de forma que podamos reducir el tiempo de ciclo y por tanto optimizar el rendimiento de la instalación. Para ello vamos a pasar a definir los tipos de direcciones que existen y que tipo de información son capaces de gestionar y transmitir. Además de describir a qué tipo de elemento está asociada cada dirección.

Lo primero que cabe destacar es la nomenclatura a seguir para definir las direcciones. Todas empiezan por el símbolo % seguidas de una letra inicial según al tipo de dirección que vayamos a gestionar. Todos los tipos de variables están definidas según la **norma IEC 61131-3**.

Estas son los distintos tipos de direcciones del autómatas y su descripción: ^[11]

INDICADOR	DESCRIPCIÓN
%I	Indica un objeto de lenguaje de entrada binario.
%W	Indica un objeto de lenguaje de entrada analógico.
%KW	Indica un objeto de lenguaje de palabra constante.
%M	Indica un objeto de lenguaje de bit de memoria.
%MW	Indica un objeto de lenguaje de palabra de memoria.
%Q	Indica un objeto de lenguaje de salida binaria.
%QW	Indica un objeto de lenguaje de salida analógica.

TABLA 2. Tipos de direcciones.

Analizando los tipos de variables podemos ver que hay dos grandes grupos. Uno en el que las direcciones son puramente digitales, sólo pueden tener valores 1 o 0; y otro grupo que son palabras de longitud 16 bits, que pueden expresar por ejemplo una entrada analógica.

2.3.1.3 Bits y palabras de sistema.

En los autómatas Modicon M340, Premium, Atrium y Quantum existen unos bits y palabras del sistema, con los que se pueden influir en su funcionamiento o realizar tareas de depuración. El usuario puede ver la evolución de estos bits durante la ejecución del programa para controlar la evolución. Algunos han de ser modificados por el usuario para devolverlos a su estado inicial, sin embargo, muchos de ellos vuelven a su estado normal a través del sistema.

Estos bits del sistema tienen la nomenclatura %Si y van desde el %S0 hasta %S123. Cada uno con un significado que viene expresado en el Manual de Unity Pro. Por ejemplo: %S0 que corresponde al arranque en frío (Este bit se define en 1 durante el primer ciclo completo de restauración del PLC en modalidad RUN o STOP. El sistema lo restablece en 0 antes del ciclo siguiente); %S4, %S5, %S6 %S7, temporizadores internos que regulan el cambio de estado de ese bit con un periodo diferente (10ms, 100ms, 1s y 1min respectivamente); entre otros muchos. Además al igual que bits de sistema, también existen palabras del sistema; su nomenclatura es %SWi y al igual que los bits tienen diversos usos.

De entre todos los bits y palabras del sistema, son especialmente interesantes a aquellos que hacen referencia a la funcionalidad de *Watchdog*. Esta funcionalidad permite mantener bajo control un sistema, en caso de fallo del programa. Este concepto, exportado de los microprocesadores, cuenta una serie de ciclos esperando que el programa escriba en un espacio de memoria dedicado a ello, de no producirse el *Watchdog* se activa y reinicia el programa; si se produce la escritura en memoria no sucede nada. Así sucesivamente el *Watchdog* vigila que el programa se esté ejecutando siempre correctamente. Además también se puede vigilar que una entrada o una variable estén entre un determinado rango, de salir de este rango se pueden asociar acciones elegidas por el programador; esto puede ser muy útil a la hora de controlar entradas analógicas al sistema.

2.3.2 Simulador del Autómata.

Como ya se ha comentado, una nueva novedad dentro del software de programación de Schneider Electric es la integración de un simulador del autómata integrado, el cual reproduce fielmente el comportamiento del programa en el PC. Todas las herramientas de puesta a punto se pueden utilizar en simulación, para aumentar la calidad antes de la instalación. Por tanto gracias a este simulador podemos llevar el código pre-depurado a la instalación, con lo cual reducimos el tiempo de parada del proceso.

- Ejecución del programa paso a paso.
- Punto de parada y de visualización
- Animaciones dinámicas para visualizar el estado de las variables y la lógica que está ejecutando.

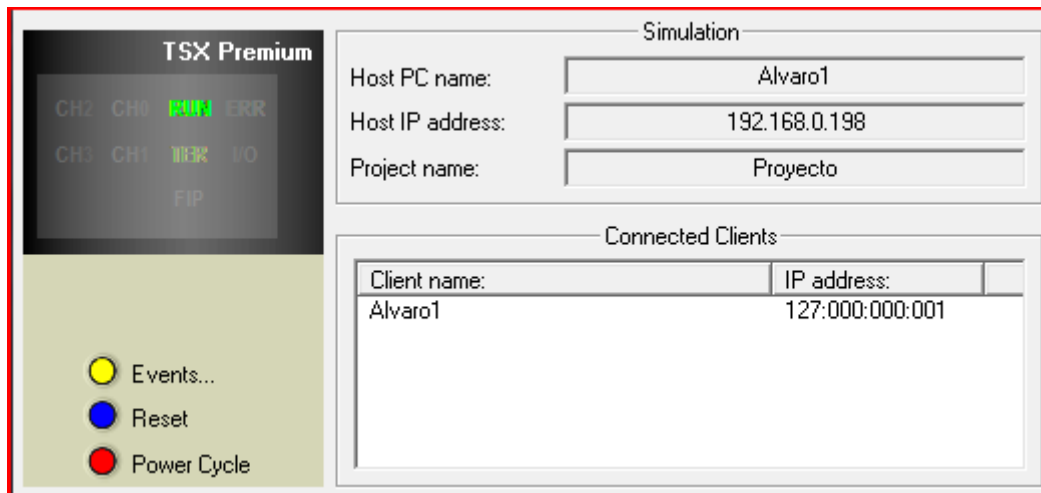


Figura 12. Simulación virtual del PLC.

Su manejo es muy sencillo, cuenta con un botón rojo de encender o pagar y uno azul para reiniciar el simulador de autómata. Si todo está correcto aparecerá iluminado *RUN*. En la parte superior derecha hace referencia a los datos del proyecto cargado junto con la dirección IP del anfitrión. Por último aparecen los clientes conectados al simulador.

2.3.3 Editor de ventanas de usuario.

Junto a la herramienta de simulación otra de las funcionalidades que más provecho se le ha sacado en este provecho es a las pantallas de usuario. Estas están compuestas por gran cantidad de información (variables dinámicas, vistas generales, textos descriptivos, etc.) y permiten vigilar y modificar rápida y fácilmente las variables de automatización. Ventana de usuario:

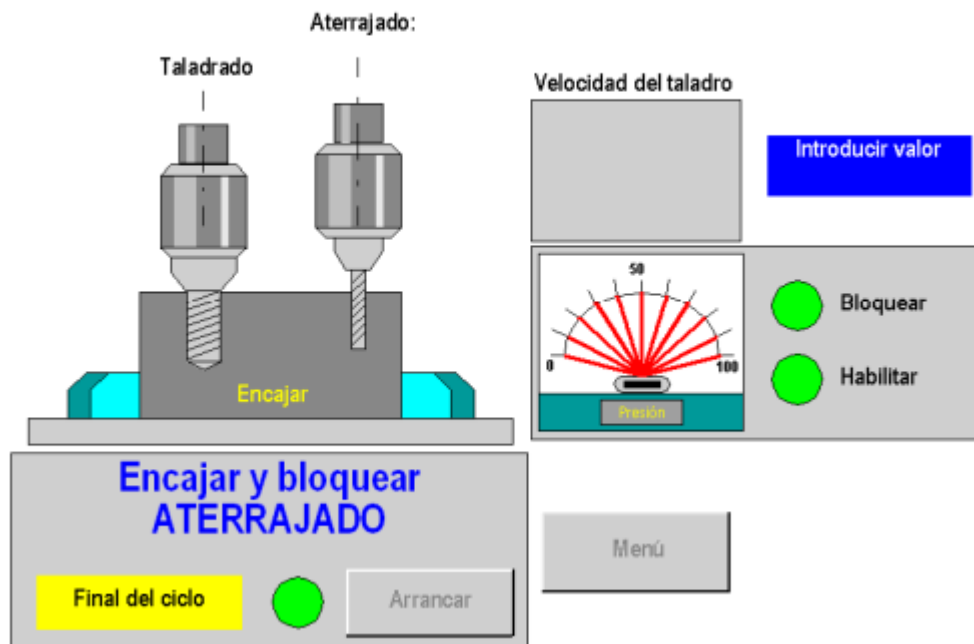


Figura 13. Ejemplo de ventana de usuario.

El editor de ventanas de usuario ofrece las siguientes prestaciones:

- Amplias funciones de visualización.
 - Elementos geométricos.
 - Línea, rectángulo, elipse, curva, polígono, mapa de bits, texto.
 - Elementos de control.
 - Botón, casilla de verificación, barra de movimiento, navegación por la pantalla, hipervínculo, campo de entrada, control numérico.
 - Elementos de animación.
 - Diagrama de barras, diagrama de tendencias, cuadro de diálogo, aparecer, desaparecer, colores intermitentes, animación de variables.
- Creación de una biblioteca para la gestión de objetos gráficos.
- Copiado de objetos.
- Creación de una lista con todas las variables utilizadas en la ventana de usuario.
- Creación de mensajes que se vayan a utilizar en las ventanas de usuario.
- Acceso directo desde las ventanas de usuario a la tabla de animación o a la tabla de referencias cruzadas para una o varias variables.

- Información sobre herramientas (Quickinfo) que ofrecen información sobre las variables.
- Gestión de ventanas de usuario en familias.
- Importación/exportación de ventanas de usuario individuales o familias enteras.

Todas estas características permiten al usuario tanto ir viendo como se desenvuelve la simulación, como influir en ella activando y desactivando salidas siendo muy útil si se quiere conseguir un programa robusto dado que se podría testar entero antes de implementarse en el autómata, posteriormente en el **CAPÍTULO 4** se explicará con más detalle la creación de objetos en estas ventanas de usuario.

CAPÍTULO III – Ejemplo desarrollado.

3.1 Enunciado del problema propuesto.

El objetivo es automatizar un proceso de embotellado, una vez que se inicia el proceso, las botellas limpias acceden a una cinta transportadora que las va llevando a las diferentes zonas: llenado, taponado, etiquetado y control de calidad. Dependiendo del resultado de este control de calidad las botellas son llevadas una zona de empaquetado o bien a una zona de desecho. Además se debe controlar el proceso de vaciado y llenado del tanque de manera independiente.

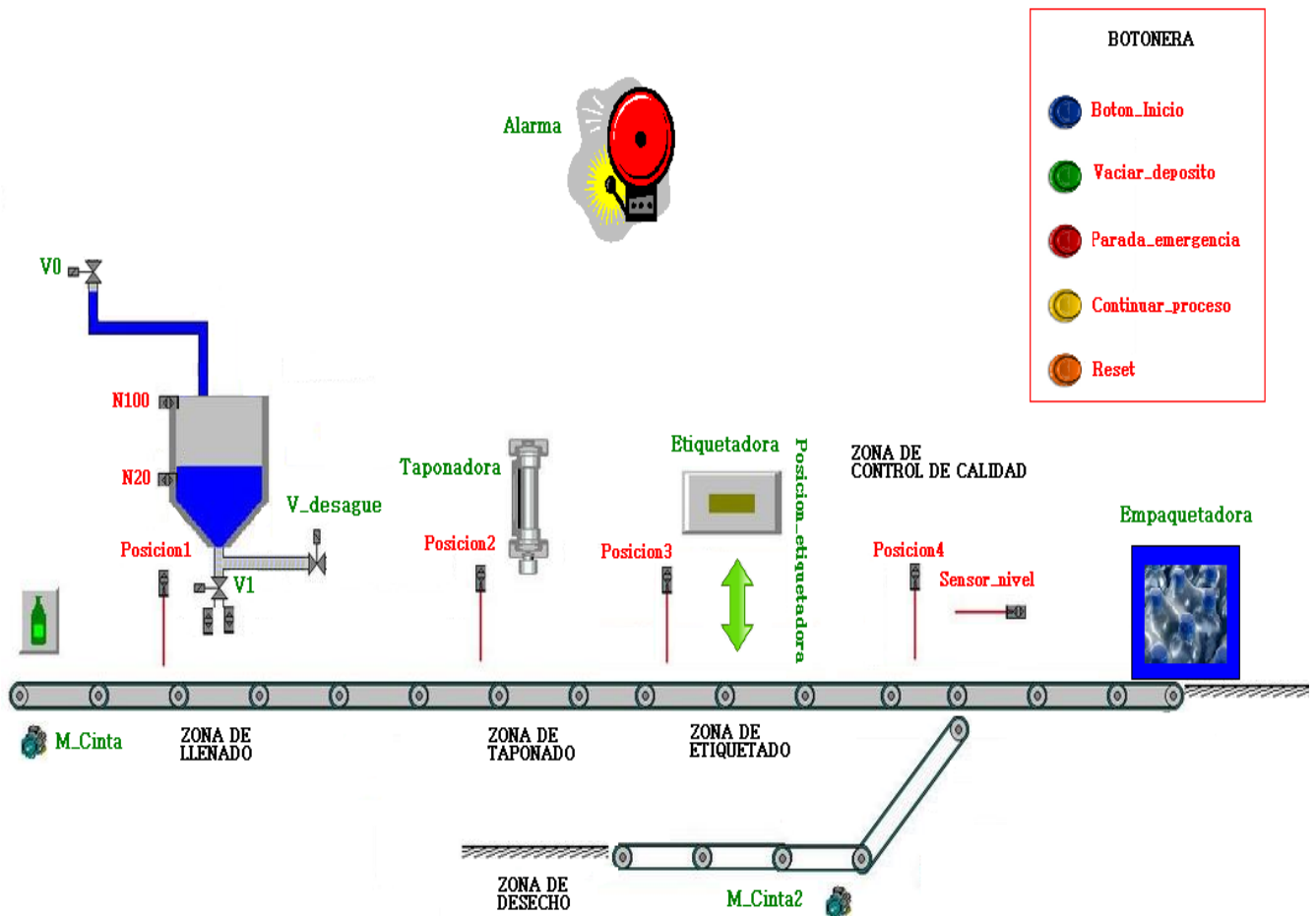


Figura 14. Esquema gráfico de la instalación.

PROCESO DE EMBOTELLADO.

- Comienzo del proceso.

El proceso sólo comenzará si no se está pulsado el botón para el vaciado del depósito (*Vaciar_depósito*), ni se ha activado la parada de emergencia (*Parada_emergencia*); además el nivel del depósito (*Nivel_deposito*) deberá ser igual o superior al 20%. Si se dan todas estas condiciones y el operario pulsa el botón de inicio (*Boton_Inicio*) se activará la cinta transportadora.

- **Proceso de llenado.**

Una vez la botella este en la zona de llenado, indicada por el correspondiente sensor de posición (*Posicion1*), se abrirá la válvula que deja pasar el contenido del tanque para llenar la botella (*V1*). El tiempo de llenado será de 5 segundos y la capacidad del tanque decrecerá un 5%, tras lo cual volverá a activarse la cinta transportadora.

- **Proceso de taponado.**

La llegada a la zona de taponado será indicada de nuevo por otro sensor de posición colocado convenientemente (*Posicion2*). La máquina encargada de taponar las botellas (Taponadora) hará su cometido en un tiempo de 8 segundos y la botella saldrá de esta zona.

- **Proceso de etiquetado.**

Cuando la botella este convenientemente colocada, detectado por el sensor correspondiente (*Posicion3*), se deberá esperar 2 segundos y ordenar que se baje la etiquetadora a la posición correcta activando la salida correspondiente (*Posición_etiquetadora*), esta salida baja el brazo en un tiempo de 2 segundos para colocar la etiqueta a la altura deseada. Tras lo cual se activará la etiquetadora (*Etiquetadora*) durante 10 segundos (en total el proceso desde que llega la botella hasta que está etiquetada abarca 14 segundos), transcurrido el tiempo se desactivan ambas salidas para dejar lista la máquina para el siguiente proceso.

- **Proceso de control de calidad.**

De nuevo un sensor de presencia (*Posicion4*) detecta que la botella ha llegado a esta zona, la cinta se para y mediante un sensor óptico (*Sensor_nivel*) situado a la altura conveniente se analiza si la botella contiene un nivel de líquido adecuado o no. Esto se hará automáticamente y sólo será necesario mantener la botella en esta posición durante 8 segundos.

- **Proceso de empaquetamiento.**

Si la botella correspondiente ha superado el control de calidad (*Sensor_nivel* = '1'), será llevada por la cinta transportadora hasta una zona almacén de empaquetamiento en un tiempo de 10 segundos. El sistema contará el número de botellas mediante un contador (*Contador_botellas*) y activará el sistema de empaquetado (*Empaquetadora*) en caso de llegar al número de botellas por paquete deseado, fijado en tres por defecto. De darse la condición, la máquina empaquetará los tres envases citados en un tiempo de 5 segundos, tras lo cual reiniciará el contador para las botellas y aumentará la variable que lleva la cuenta del número de paquetes elaborados (*Contador_paquetes*); si todavía no se ha alcanzado la cantidad necesaria el proceso seguirá con su curso, es decir, volverá al principio.

- **Proceso de botellas defectuosas.**

En el caso de que en la etapa de control de calidad se determine que el envase no tiene el nivel de líquido adecuado (*Sensor_nivel* = '0'), se activará durante 10 segundos una cinta transportadora diferente (*M_Cinta2*) que llevará la botella defectuosa a otra zona

determinada para elementos determinados como no válidos. Se deberá llevar la cuenta de envases defectuosos detectados por el control de calidad (*Contador_botellas2*), ya que cuando se acumulen tres botellas defectuosas en la zona de desechos se debe parar el proceso y activar una alarma (*Alarma*). Para continuar con el proceso deben haber pasado 10 segundos tras la activación de la alarma y que un operario pulse el botón de continuar con el proceso (*Continuar_proceso*) indicando que ha retirado los envases defectuosos y ha revisado que no hay ningún problema adicional. Tras lo cual el proceso volverá a su comienzo, al igual que ocurriría si no se ha alcanzado el límite de tres botellas desechadas.

- **Transporte de botellas.**

Para llevar las botellas de una fase a otra, se utiliza una cinta transportadora movida por un motor de corriente continua (*M_Cinta*), en todas las etapas la cinta se activará, llevará la botella hacia la siguiente fase y se parará cuando el correspondiente sensor de presencia lo detecte, excepto en el caso de que la botella haya sido declarada óptima tras pasar por todas las fases; en este caso se activará durante un intervalo de 8 segundos que es el tiempo que tarda en llegar a la zona de empaquetado.

Además habrá una segunda cinta transportadora (*M_Cinta2*) encargada de llevar las botellas calificadas como defectuosas a la zona indicada en un tiempo de 10 segundos.

CONTROL DEL NIVEL DEL TANQUE.

- **Llenado del depósito.**

Para que el tanque se comience a llenar se ha de dar las condiciones de que no se encuentre activado el botón de vaciado (*Vaciar_deposito*), además sólo se rellenará cuando esté por debajo de un 20% de su capacidad; comprobándose de dos formas: por un sensor de ultra sonidos colocado a esa altura (*N20*) que cuando este a 0 indicará que el nivel está por debajo, por otro lado se debe llevar la cuenta del porcentaje con una variable (*Nivel_deposito*). Si cualquiera indica que está por debajo de la capacidad indicada se procederá a llenar abriéndose la válvula de entrada de fluido (*V0*). El proceso de llenado será un bucle cerrado en el que la variable del nivel del depósito se irá incrementando un 5% cada 2 segundos hasta llegar al 100% indicado únicamente por otro sensor de ultrasonidos colocado en la parte superior del tanque (*N100*), volviendo así a su estado de reposo.

- **Vaciado del depósito.**

Las condiciones que se han de dar para que se inicie el proceso de vaciado son dos: que se pulse el botón para vaciarlo (*Vaciar_deposito*) y que la capacidad del tanque (*Nivel_deposito*) se encuentre por encima de 0. Si se cumplen ambas imposiciones se abrirá la válvula del desagüe (*V_desague*) y comenzará un bucle cerrado en el que el depósito irá decrementando su capacidad de 5 en 5% cada 2 segundos hasta quedar completamente vacío y volver a su posición de reposo.

PARADA DE EMERGENCIA.

Como en todo sistema de producción, se programará una parada de emergencia. Esta debe ser accesible en cualquier estado. Se resetearán todas las salidas excepto *Posicion_etiquetadora* y se interrumpirá cualquier transición. Además si un proceso se queda a medias cuando se reanude se debe continuar exactamente por donde iba controlándose que se cumplan los tiempos establecidos para cada uno. Se activará al pulsar la seta de parada de emergencia (*Parada_emergencia*) y se desactivará al volver a su posición original esta misma señal. Mientras el sistema permanezca parado se ha de notificar mediante una alarma (*Alarma*).

RESET DE CONTADORES.

El sistema cuenta con varios contadores, paquetes producidos (*Contador_paquetes*), botellas pendientes de empaquetar (*Contador_botellas*) y el de las botellas defectuosas almacenadas (*Contador_botellas2*); se podrá hacer un reset de todos ellos con pulsar el botón de Reset en el panel de mandos (*Reset*), todos los valores se inicializarán independientemente del estado en el que nos encontremos.

3.2 Variables del proceso.

Una vez enunciado el problema, se han dispuesto una serie de variables que llevarán a conseguir el objetivo marcado. Dividiremos estas variables en tres grupos: aquellas que son entradas o salidas (todas ellas serán del tipo EBOOL) y llevan asociada una dirección (para actuar en modo simulación se quitarán las direcciones, dado que en determinadas versiones de Unity Pro da errores), variables creadas por el programa de manera automáticas y el resto de variables.

3.2.1 Variables de entrada y salida.

- ENTRADAS.

Nombre	Dirección	Comentario
Reset	%I0.1.0	Botón RESET, para reiniciar los contadores de las variables
Boton_inicio	%I0.1.1	Botón para iniciar el proceso.
Continuar_proceso	%I0.1.2	Botón que se pulsará para indicar que se ha subsanado cualquier error.
Parada_emergencia	%I0.1.3	Pulsador para realizar una parada de emergencia.
Vaciar_deposito	%I0.1.4	Botón utilizado para vaciar el contenido del tanque.
Posicion1	%I0.1.5	Sensor que indica que la botella ha llegado a la zona de llenado.
Posicion2	%I0.1.6	Sensor que indica que la botella ha llegado a la zona de taponado.
Posicion3	%I0.1.7	Sensor que indica que la botella ha llegado a la zona de etiquetado.
Posicion4	%I0.1.8	Sensor que indica que la botella ha llegado a la zona de control de calidad.
N20	%I0.1.9	Sensor que indica que el nivel del tanque está por encima del 20%.
N100	%I0.1.10	Sensor de nivel que indica que el tanque está al 100%, lleno.
Sensor_nivel	%I0.1.11	Sensor que indica si la botella está rellena o no.

TABLA 3. Variables de Entrada.

- SALIDAS.

Nombre	Dirección	Comentario
V0	%Q0.2.0	Válvula que abre la entrada de líquido al depósito.
V1	%Q0.2.1	Válvula que abre el tanque para llenar una botella.
V_desague	%Q0.2.2	Válvula utilizada para vaciar el depósito.
M_Cinta	%Q0.2.3	Motor para mover la cinta transportadora principal.
M_Cinta2	%Q0.2.4	Motor para encender la cinta transportadora de botellas desechadas.
Taponadora	%Q0.2.5	Activa la máquina encargada de taponar las botellas.
Maquina_etiquetado	%Q0.2.6	Activa el mecanismo encargado de etiquetar las botellas.
Posicion_etiquetadora	%Q0.2.7	Motor para mover la máquina de etiquetado. 1 abajo, 0 arriba.
Empaquetadora	%Q0.2.8	Máquina encargada del empaquetado de botellas.
Alarma	%Q0.2.9	Señal acústica que indica que hay algún problema.

TABLA 4. Variables de salida.

3.2.3 Variables generadas por el programa.

A medida que vamos generando etapas de nuestro SFC y transiciones, van apareciendo en el programa variables que están asociadas a ellas. Aparecen con un candado a la izquierda que indica que están protegidas y no se pueden modificar por el usuario, solo consultar su valor.

- VARIABLES DE TRANSICIONES.

Nombre	Tipo	Dirección	Valor	Comentario
Vacio	BOOL			
Tiempo_tapon	BOOL			
Tiempo_Llenado	BOOL			
Tiempo_Etiquetado	BOOL			
Tiempo_Empaquetado	BOOL			
Tiempo_desecho	BOOL			
Tanque_vacio	BOOL			
Problema_solucionado	BOOL			
P4	BOOL			
P3	BOOL			
P2	BOOL			
P1	BOOL			
Optima	BOOL			
No_vacio	BOOL			
No_Lleno	BOOL			
Lleno	BOOL			
Empaquetado_SI	BOOL			
Empaquetado_NO	BOOL			
Desague	BOOL			
Dep_des_lleno	BOOL			
Control_Defectuoso	BOOL			
B_inicio	BOOL			

TABLA 5. Variables de transiciones.

- VARIABLES DE ESTADO.

Nombre	Tipo	Dirección	Valor	Comentario
Vaciado	SFCSTEP_STATE			
Tapa	SFCSTEP_STATE			
Reposo_Tanque	SFCSTEP_STATE			
Rellenar	SFCSTEP_STATE			
Llenado	SFCSTEP_STATE			
Inicio	SFCSTEP_STATE			
Etiquetado	SFCSTEP_STATE			
Empaquetado	SFCSTEP_STATE			
Desecho	SFCSTEP_STATE			
Deposito_des_Lleno	SFCSTEP_STATE			
Control_calidad	SFCSTEP_STATE			
Cinta_5	SFCSTEP_STATE			
Cinta_4	SFCSTEP_STATE			
Cinta_3	SFCSTEP_STATE			
Cinta_2	SFCSTEP_STATE			
Cinta_1	SFCSTEP_STATE			
Embotelladora	SFCCHART_STATE			
Control_deposito	SFCCHART_STATE			

TABLA 6. Variables de Estado.

3.2.3 Otras variables.

NOMBRE	TIPO	COMENTARIO
NOT_Parada_emergencia	BOOL	NOT(Parada_Emergencia). Condición para que se produzcan las transiciones en los SFC.
Contador_botellas	INT	Cuenta las botellas llenadas correctamente.
Contador_botellas2	INT	Cuenta el número de botellas desechadas.
Contador_Paquetes	INT	Cuenta el número de paquetes que se han realizado hasta el momento.
Nivel_deposito	INT	Indica el nivel del líquido en el depósito.
Posicion	INT	Indicará el estado de la botella y la representación que ha de aparecer.
Tiempo_etiquetadora	INT	Muestra el tiempo que resta para que acabe el proceso de etiquetado.
Tiempo_etiquetadora2	INT	Almacena como entero el tiempo del estado de etiquetado.
T_aux2	INT	Almacena como entero el tiempo de T_aux.
T_aux	TIME	Guarda el tiempo que está activa la parada de emergencia.
Auxiliar	INT	Variable usada sólo en el Modo Docente para almacenar tiempo.
Deposito_aux	INT	Variable usada sólo en Modo Docente para controlar el llenado y vaciado del tanque.

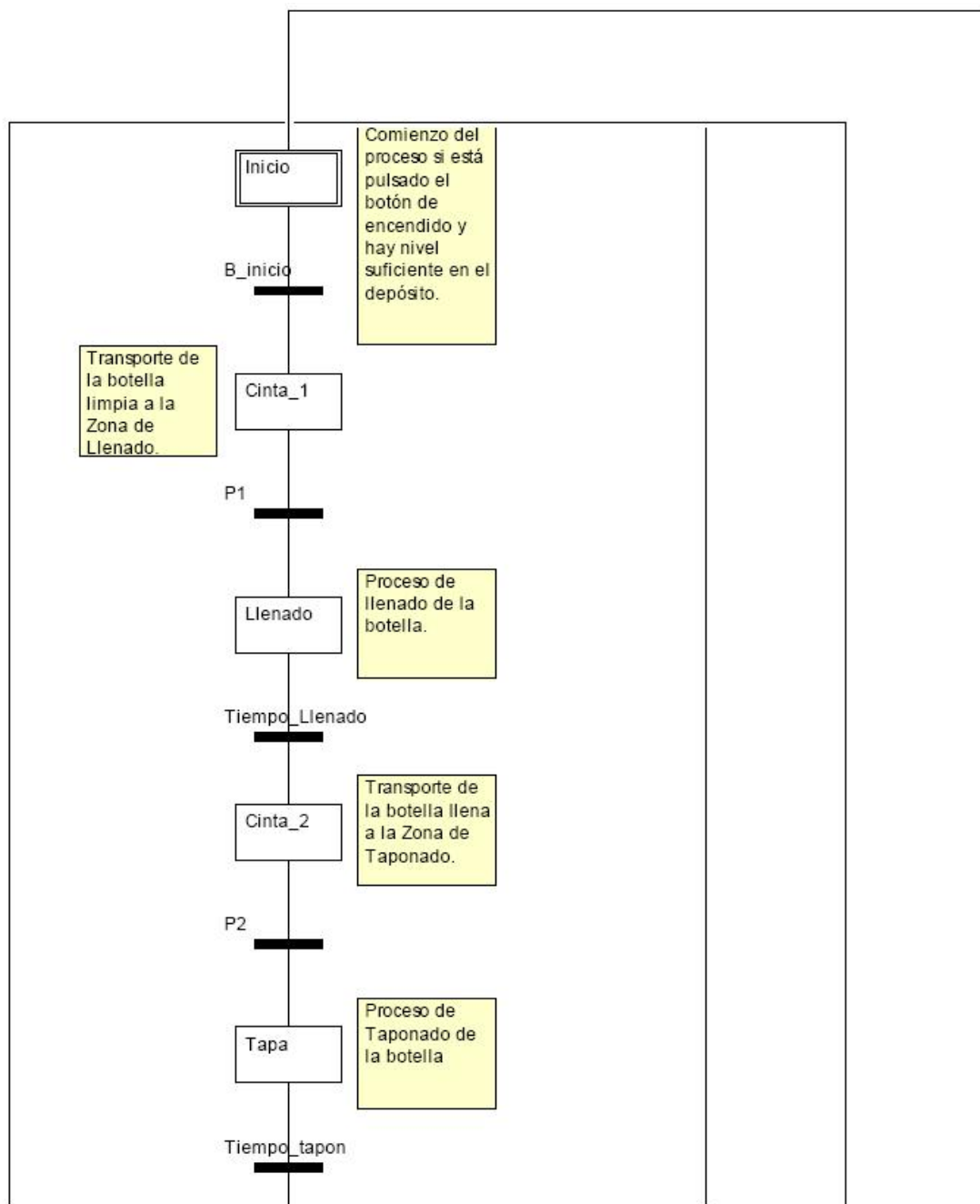
TABLA 7. Otras variables.

3.3 Secciones del programa.

Para desarrollar el problema se han propuesto cuatro secciones: dos diagramas SFC programados en LD, uno para el proceso de embotellado y otro para el control del depósito; y dos hojas de LD, una hoja de LD en la que están las condiciones que desarrollan la parada y otra para el reseteo de contadores.

3.3.1 Embotelladora.

Esta parte del programa es la principal, se encarga de recibir las botellas limpias y pasarlas por la zona de llenado, las taponas, las etiqueta, pasa el control de calidad y decide si las botellas son aptas para empaquetarlas o desecharlas. Presenta el siguiente diagrama SFC:



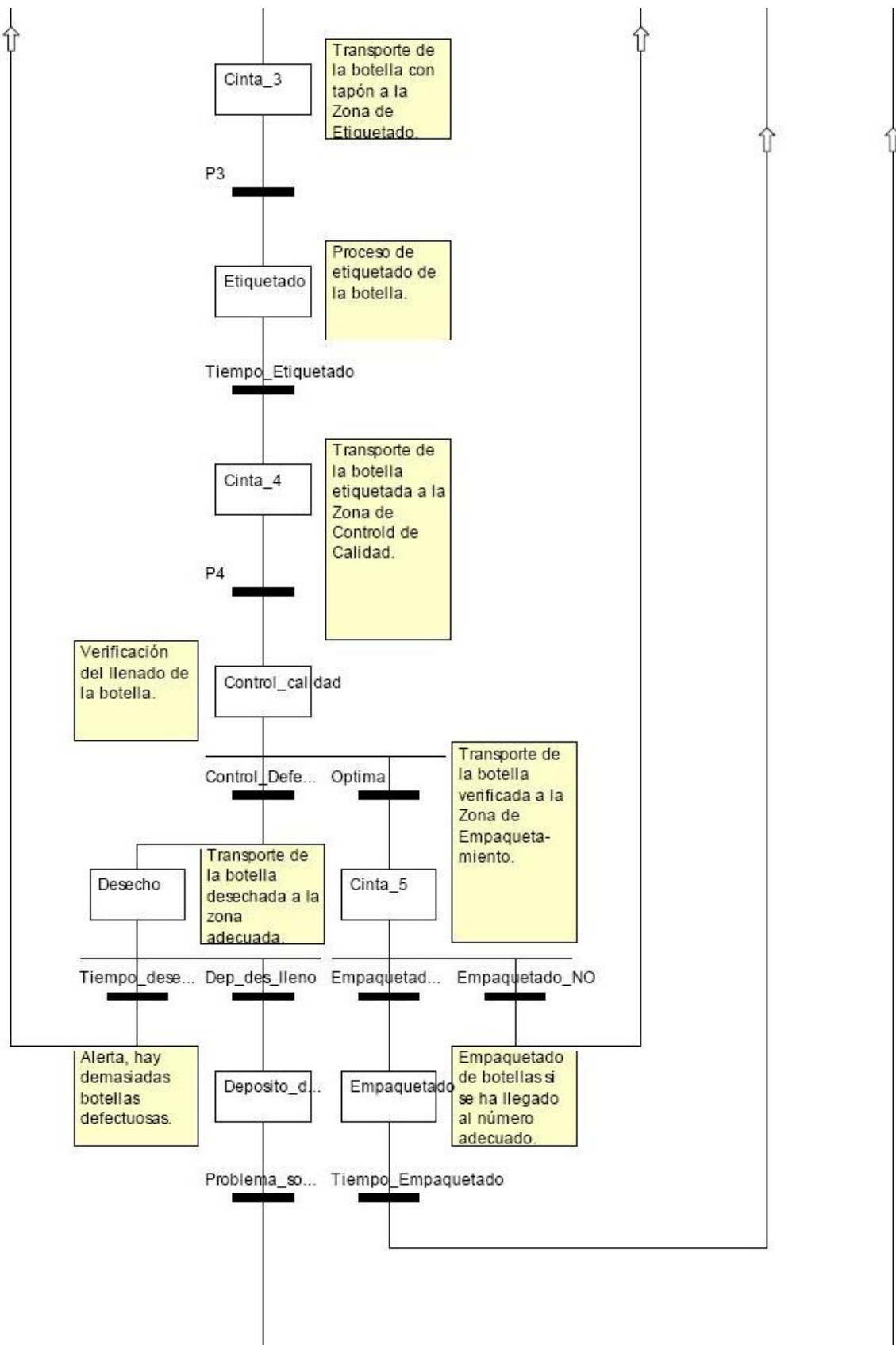


Figura 15. Diagrama SFC Embotelladora.

3.3.2 Control del depósito.

Esta sección se encarga de todo lo relativo al tanque en el que se deposita el líquido que va en el interior de las botellas, controla su nivel impidiendo que baje del 20% y se encarga de vaciar el depósito si así se desea.

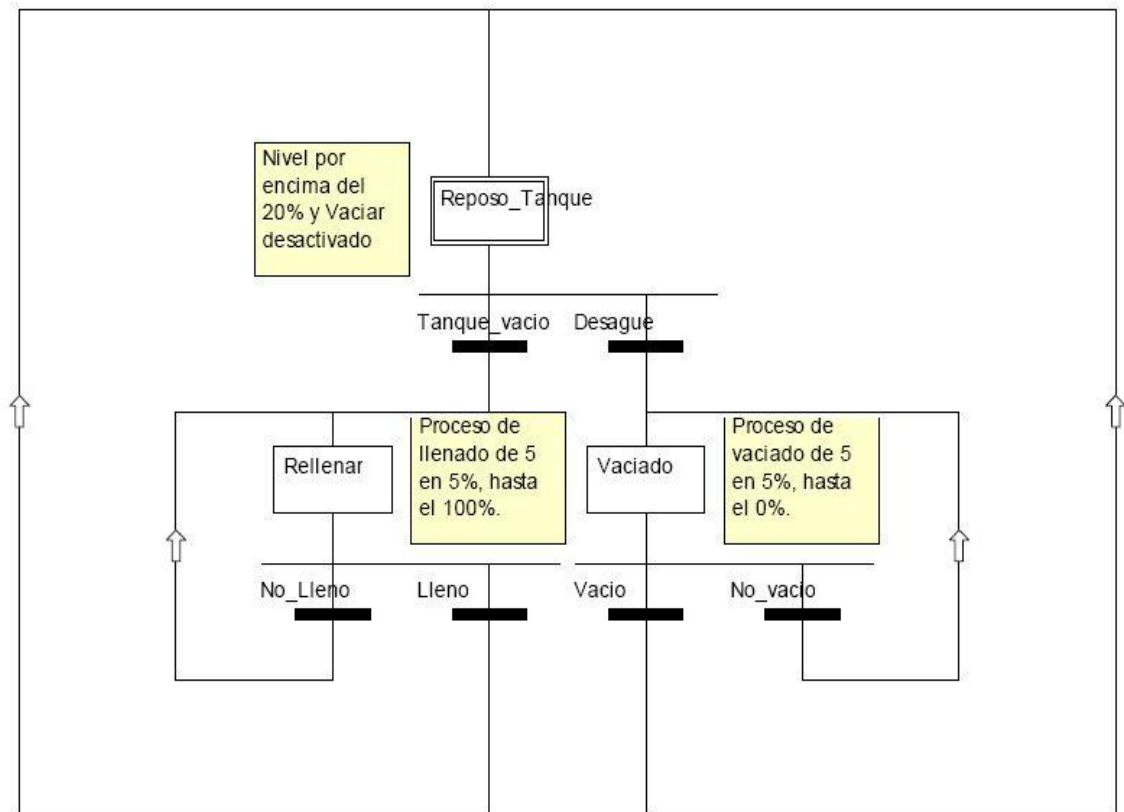


Figura 16. Diagrama SFC Control depósito.

3.3.3 Parada de emergencia.

Esta sección se ocupa de llevar a cabo la parada de emergencia, siendo accesible desde cualquier parte del proceso, de ahí que se desarrolle en una hoja de contactos. Dado que para esta parte no se ha diseñado en GRAFCET un diagrama sería el siguiente:

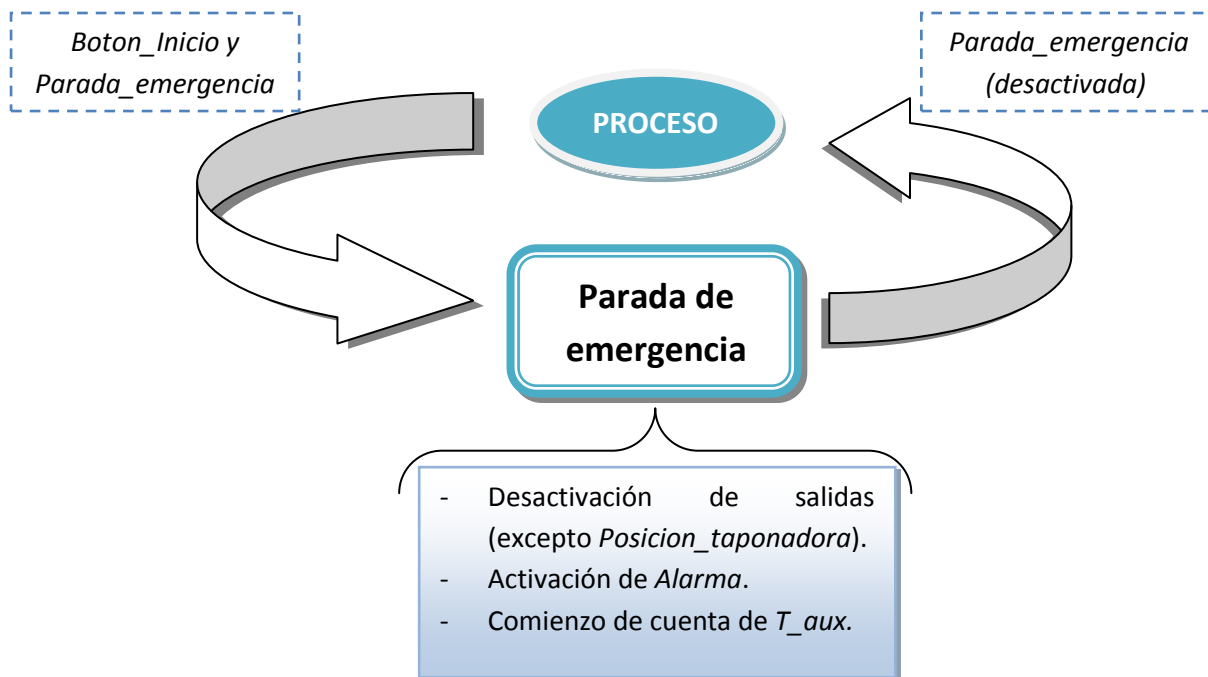


Figura 17. Diagrama Parada de Emergencia.

3.3.4 Reinicio de contadores.

La sección más simple de todas, realizada en una hoja de contactos para que sea siempre accesible. Con pulsar el Botón *Reset* se llevará a 0 el valor de la cuenta de todos los contadores del programa.

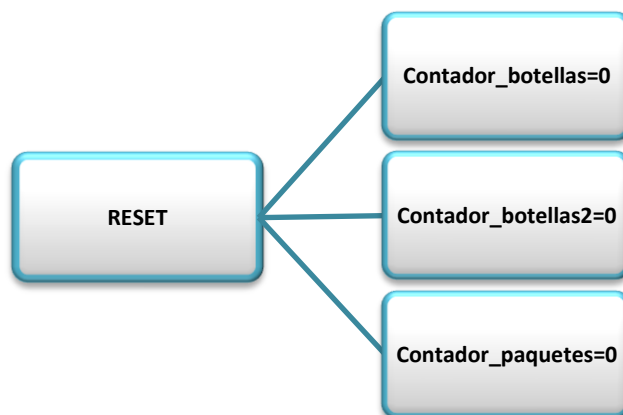


Figura 18. Diagrama reset de contadores.

3.4 Programación.

Se ha descrito que el programa tiene 4 secciones, dos configuradas totalmente en lenguaje de contactos y otras dos en SFC, pero en el interior de dichos diagramas también hay pequeñas partes (transiciones, activaciones...) que van en lenguaje de contactos o que llevan bloques (como comparación u operación) programados en ST. Todo ello se debe a la gran versatilidad que ofrece Unity Pro para combinar todas sus herramientas sacando el mayor rendimiento posible.

En ambos GRAFCET se subdividen en etapas, y cada etapa tiene en su interior una serie de acciones que llevan asociadas un descriptor. Hay diversas clases de descriptores, pero los utilizados en este proyecto son los siguientes:

- **N:** Se ejecuta la acción programada en la sección (o pone a “1” la variable asignada) durante el tiempo que está activa la etapa.
- **P1:** Se ejecuta la acción programada en la sección (o pone a “1” la variable asignada) cuando la etapa pasa de inactiva a activa (acción al activar).
- **P0:** Se ejecuta la acción programada en la sección (o pone a “1” la variable asignada) cuando la etapa pasa de activa a inactiva (acción al desactivar).
- **S:** Se encarga de poner a “1” la variable asignada.
- **R:** Se encarga de poner a “0” la variable asignada.

Toda la programación detallada se adjuntará en el *Anexo II – Documentación Técnica del proyecto*, pero en este apartado se pretenden explicar algunos matices de las soluciones que se han adoptado en aquellas zonas que se ha encontrado una mayor dificultad. Se proceden a explicar las siguientes configuraciones:

- **Control de las diferentes figuras de la botella.**

La botella va cambiando su apariencia según va avanzando el proceso, esto se debe a ir modificando el valor de la variable *posición* mediante un bloque *Operate* dando el valor adecuado en el momento correcto.

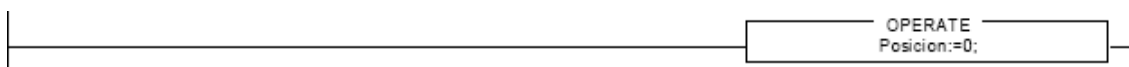


Figura 19. Programación posición botellas.

Además para que no surjan saltos de la botella de un lugar a otro, por equivocaciones en la activación de los botones, también se han incluido restricciones que impiden que la botella avance a la próxima posición si el resto de los sensores de posición, previos o posteriores, están activados, como se muestra en la *Figura 20*:



Figura 20. Programación restricciones de posición.

- Reset de contadores.

Esta parte se incluye en la hoja LD denominada *Reset_contadores* está de manera independiente en una hoja LD para que sea accesible desde cualquier parte del programa. Su función es reiniciar los contadores del programa a través de un bloque *Operate*.

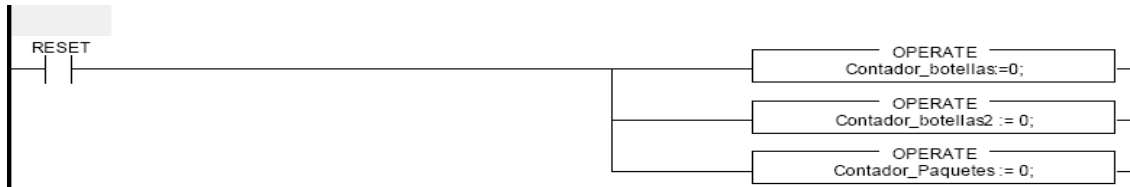


Figura 21. Programación reinicio de contadores.

- Cuenta atrás en la etiquetadora.

En la simulación podemos observar como en un campo de entrada aparece el valor del tiempo restante para que acabe el proceso. Para conseguir esto en la simulación se ha tenido en cuenta que primero se esperan dos segundos y se baja la etiquetadora que tarda otros dos segundos en colocarse en posición. Unity Pro no nos deja operar con variables de tipo *TIME* por lo que almacenamos el tiempo del estado en una variable *INT* (*Tiempo_etiquetado2*), para ello nos servimos de uno de los bloques presentes en la biblioteca de funciones:

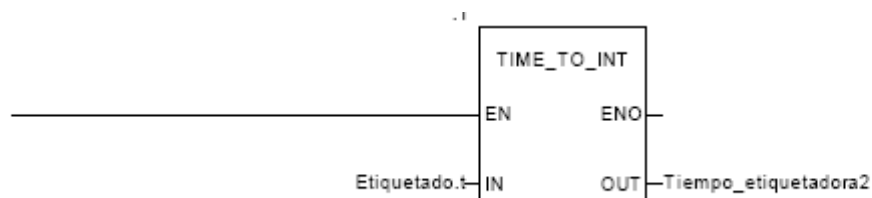


Figura 22. Conversión de TIME a INT.

Una vez que tenemos el tiempo en *INT* debemos tener en cuenta que aparecerá en ms. En primer lugar realizamos la comprobación de que estemos entre 4000 ms y 14000ms que es el tiempo en el que queremos realizar la cuenta atrás y tras ello a la variable *Tiempo_etiquetado* le asignamos el valor de la resta de 14s menos *Tiempo_etiquetado2* (transformado a segundos), esto se muestra en la *Figura 23*:

$(\text{Tiempo_etiquetadora2} - T_aux2) \geq 4000 \text{ AND } (\text{Tiempo_etiquetadora2} - T_aux2) \leq 14000$ $\text{Tiempo_etiquetadora} = 14 - ((\text{Tiempo_etiquetadora2} - T_aux2) / 1000);$

Figura 23. Programación cuenta atrás etiquetadora.

La parte que vemos de T_{aux} y T_{aux2} tiene que ver con la parada de emergencia que se explica a continuación.

- **Parada de emergencia.**

Esta es la etapa más complicada de realizar, la forma ideal de programar una parada de emergencia sería parando el sistema al desactivar el bit de estado %S30, para luego mediante un evento hacer poner a '1' el bit %S1 y hacer un rearranque en caliente. Pero los eventos sólo pueden usarse con determinadas tarjetas de conteo que los autómatas de los laboratorios no disponen. Por lo tanto no podemos parar la tarea maestra con %S30, dado que no tenemos forma de volver a activarla.

Ante esta tesitura se ha optado por una solución compuesta de tres puntos de acción:

1. Desactivar todas las salidas (menos *Posicion_etiquetadora*) y activar la alarma.
2. Por un lado impedir las transiciones cuando se active la parada de emergencia.
3. Contar el tiempo que el sistema permanece parado para realizar las comprobaciones convenientes y lograr que cada etapa este el tiempo programado.

El primer punto es el más sencillo de conseguir, se ha solucionado en una hoja de contactos y está siempre accesible desde cualquier parte del programa. Cuando se activa la variable *Parada_emergencia* se realizan todo lo comentado relativo a las salidas, además de activarse un bit de memoria %MW100.0 que nos servirá para posteriormente contar el tiempo. Tras desactivarse esta variable se resetea la alarma y el bit nombrado.

Para lograr impedir las transiciones nos servimos de una de las propiedades de Unity Pro, dentro de cada SFC podemos acceder a sus *propiedades/condición* y ligar su ejecución a que una variable se encuentre activa, como se muestra en la *Figura 24*:

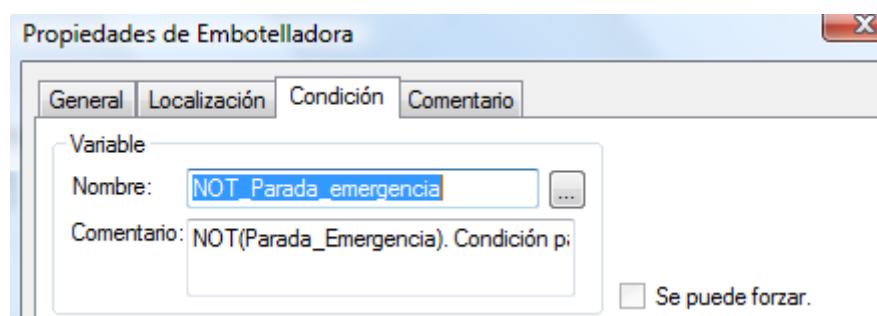


Figura 24. Configuración de condición en SFC.

Dado que queremos que el GRAFCET se active cuando no esté activada la *Parada_emergencia* y sólo podemos ligarlo a que trabaje cuando esté activada una variable, creamos un BOOL auxiliar *NOT_Parada_emergencia*, que tomará valores contrarios a *Parada_emergencia* y conseguiremos nuestro objetivo de impedir las transiciones cuando se active la parada.

El tercer punto consiste en contar el tiempo que permanece parado el sistema en caso de activarse el procedimiento de la parada y realizar las respectivas comprobaciones. Para contar el tiempo nos servimos del bit %MW100.0 y del bit del sistema %S5, que genera flancos positivos cada 100ms, para ir contando el tiempo de la siguiente forma:



Figura 25. Contaje de tiempo de parada.

De esta forma iremos guardando el tiempo en la variable T_aux de tipo TIME, que nos servirá para poder completar las condiciones de las transiciones de aquellas etapas configuradas por tiempo. Si antes en una transición teníamos la condición de cinco segundos, ahora para configurar correctamente la parada de emergencia tendremos la suma de esos cinco segundos más T_aux . De esta forma conseguimos que cuando desactivamos la parada no salte a la siguiente etapa, dado que al no poder usar eventos, el tiempo de cada estado sigue contando. En la *Figura 26* se muestra un ejemplo de lo comentado:

Cinta 5. $t \geq (t\#10s + T_aux)$

Figura 26. Programación tiempo auxiliar.

Todo lo hablado sirve para todos los casos excepto para uno, el campo de entrada de la simulación de la etiquetadora que nos muestra el tiempo que falta. Crearemos otra variable T_aux2 que guardara el valor de T_aux en un INT expresado en ms. Después al ya tenerlo en tipo INT podemos operar con ello para sumar o restar según convenga y conseguir que se mantenga intacta la cuenta atrás. Todo este ajuste viene expresado anteriormente en la *Figura 23 (Programación cuenta atrás etiquetador)*.

Por último comentar que estas consideraciones con el tiempo sólo se han tenido en cuenta para los procesos que conciernen al SFC de la embotelladora, dado que en el depósito al tener dos sensores de nivel siempre tendremos controlados los límites de llenado que nos sirven de guía.

CAPÍTULO IV – Pantalla de operador.

4.1 Definición y creación.

Como se ha comentado en puntos anteriores, Unity Pro ofrece la posibilidad de crear pantallas de operador, en ellas es posible insertar objetos como botones, indicadores, textos, números, barras, casilla de verificación e imágenes desde una librería de pantallas de operador u otra ubicación.

Su diseño es totalmente libre, no hay que seguir ninguna regla especial. Se utilizan las herramientas más tradicionales como copiar, cortar, pegar y el ratón para desplazar los objetos o acceder a las propiedades (mediante un doble clic). La librería de operador permite utilizar numerosos objetos gráficos ya creados que facilitan la creación de pantallas de operador.

A la vez que se ha ido creando el sistema y su programación, se ha ido generando una pantalla de operador en la que representar todo y ver gráficamente cómo va siendo su evolución. Esta pantalla estará operativa tanto en el *modo de simulación*, como en el *modo estándar* con el autómata. En el primer modo podremos interactuar con el programa, incidiendo en el valor de las variables con los elementos adecuados; si hemos transferido el proyecto al PLC y es éste el que está ejecutando el programa, la pantalla gráfica irá reflejando el desarrollo del proceso, pero sin posibilidad de incidir en él desde la pantalla de operador.

Para crear una pantalla de este tipo se ha de hacer un clic derecho en *pantallas de operador* desde el explorador de proyectos y seleccionar *nueva pantalla*, como se muestra en la *Figura 27*:

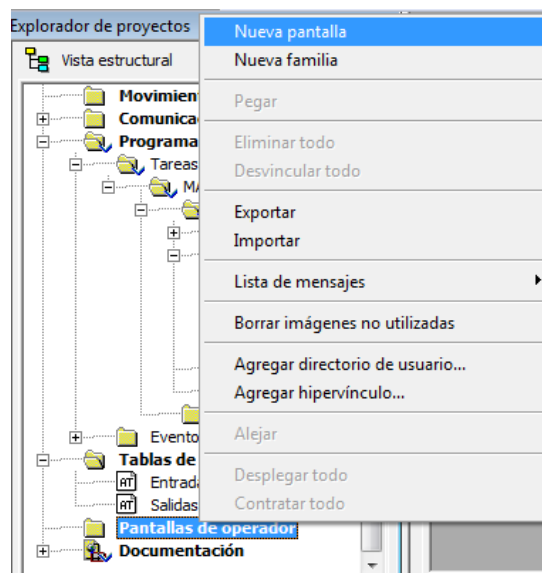


Figura 27. Creación de una pantalla de operador.

La pantalla de operación desarrollada en este trabajo es la presentada en la *Figura 28*:

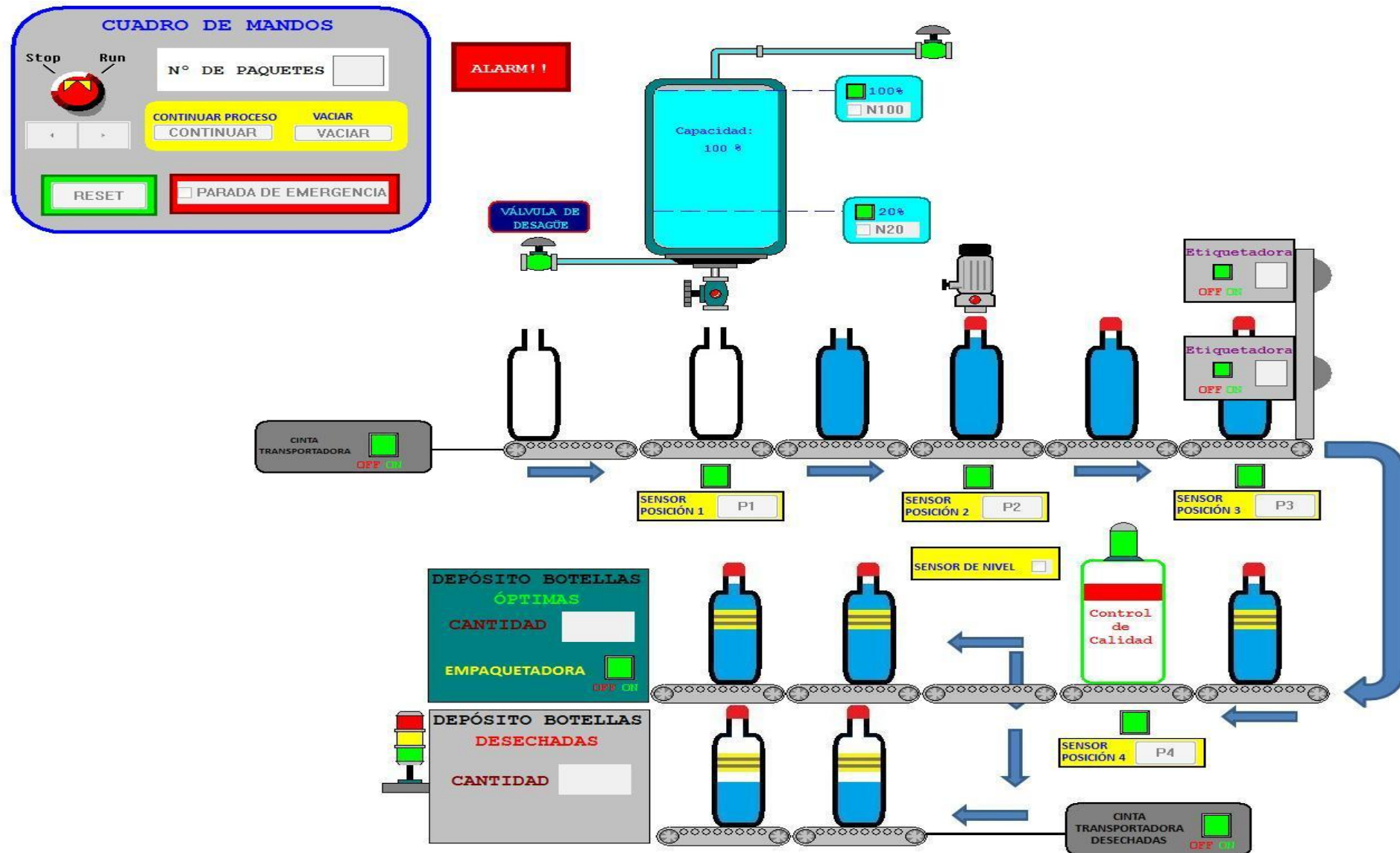


Figura 28. Gráfico de la instalación.

4.2 Explicación de elementos.

Muchos de los elementos existentes en la *Figura 28*, están presentes en la biblioteca proporcionada por el programa. Para acceder a esta biblioteca tendremos que ir a *Herramientas* y marcar la opción de *Librería de pantallas de operador*. Tras lo que aparecerá una ventana a nuestra derecha en la que podremos seleccionar diferentes elementos ya diseñados, que podrán ser dinámicos o estáticos. Las opciones que nos proporcionan se muestran en la *Figura 29*; pudiéndose desplegar cada carpeta para ver su contenido.

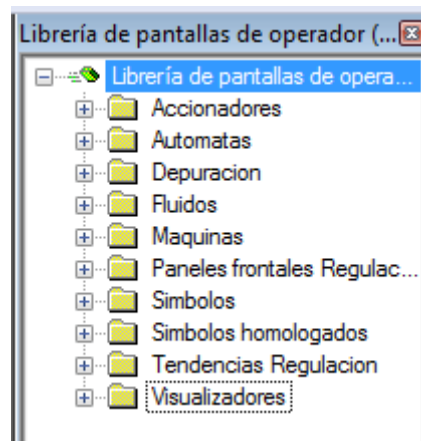


Figura 29. Librería de pantalla de operador.

Tras desplegar la carpeta deseada y se ha de buscar el componente que se adapte a nuestras necesidades y mediante un simple copia y pega lo situaremos en nuestra pantalla de operador. Tanto estos elementos como otros que creemos nosotros, los podemos clasificar en función de la variable que estén asociados (algunos no están asociados a ninguna variable y son siempre visibles) y el tipo de animación o movimiento que ofrezcan:

- **Asociados a una variable booleana.**

Al estar asociados a una variable booleana podemos hacer que aparezcan o desaparezcan según esté el valor de la variable a la que los hemos asociado. La aplicación más simple es que aparezcan cuando está activada la variable deseada y lo podemos ir haciendo más complejo como es superponer piezas del mismo tamaño y distinto color para dar más realismo. Un ejemplo sencillo es el cuadro de Alarma que aparece sólo cuando está activada la variable *Alarma*, un ejemplo más complejo pueden ser las válvulas que se ponen en verde cuando están abiertas y en caso contrario permanecen en color rojo.

- **Asociados a una variable tipo entero (INT).**

En este caso se produce la representación más dinámica y más real de todas, una forma va variando su contenido en función de la variable que se asocia. Esto sucede con el tanque que contiene el líquido con el que se llena las botellas.

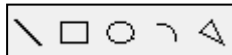
Dichos objetos pueden ser modificados y podemos combinarlos para crear otros nuevos, además podemos dibujar formas, líneas, cargar imágenes... La barra de herramientas que nos muestra todas estas posibilidades aparece en la *Figura 30*:



Figura 30. Barra de herramientas de la pantalla de operador.

Estos botones ofrecen las siguientes operaciones, que posteriormente uniremos unas a otras para explicar detalladamente cómo se han ido formando elementos presentes en el gráfico de la instalación.

- **Dibujar trazos o áreas.**



Cada uno de los elementos permitirá dibujar en la pantalla el elemento que indica su dibujo: una línea, un cuadrado, una elipse, un círculo, un arco o un polígono. A partir de ellos podremos ir componiendo de forma sencilla un elemento mucho mayor.

- **Insertar cuadro de texto.**



Esta opción nos permitirá introducir un campo de texto, fijo que no dependa de ninguna variable.

- **Insertar imagen.**



Podremos insertar una imagen en la pantalla de operador desde cualquier ubicación de nuestro ordenador, que hubiéramos guardado previamente. También es posible si tenemos una planta de producción fotografiarla para hacer sobre él las animaciones.

- **Examinar pantalla.**



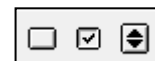
Será útil si trabajamos con varias pantallas de operador para pasar de una a otra pantalla.

- **Campo de entrada.**




Permitirá visualizar el valor que va tomando una variable a lo largo de la ejecución del programa.


- **Botón de comando, casilla de verificación y cuadro de giro.**

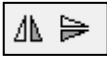


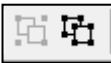
Estos elementos permitirán ir modificando el valor de una variable de tipo BOOL o EBOOL a lo largo del modo simulación, según se active o desactive (en el caso de botón de comando y la casilla de verificación) o se seleccione una de las dos opciones disponibles (en el cuadro de giro), variaremos de '1' a '0' y viceversa el valor de la variable a la que la hemos asociado. Además el botón de comando existe una opción

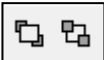
para activar la retención, para poder usarlo como un simple pulsador o bien un pulsador con retención.

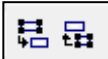
- **Indicador de escala.** 


Una vez insertado en la pantalla de operador, este indicador nos permitirá influir sobre el valor de una variable en una escala seleccionada, según se desplace a un lado u a otro el indicador.
- **Intercambio explícito.** 

Gracias a este operador será posible realizar un intercambio de información mediante de operaciones de leer, escribir o guardar datos. Todo gracias al tratamiento de información en buses de datos configurando las direcciones de manera adecuada.
- **Girar horizontalmente y verticalmente.** 

Con estos elementos podremos obtener la imagen simétrica del objeto que seleccionemos.
- **Agrupar y desagrupar.** 

Una de las opciones más útiles y más utilizadas, está presente en muchos programas que utilizamos habitualmente como Microsoft Word, nos permite juntar varios objetos, inicialmente separados, para juntarlos y tratarlos como un conjunto. A su vez también podemos conseguir el efecto contrario, separar un conjunto que originalmente se nos presenta como agrupado.
- **Primer/Segundo plano.** 

Cuando superponemos un objeto a otro podemos elegir cuál de los dos queremos que quede por encima, es decir, que tape al otro y que se vea. Esto es lo que se consigue con estos dos botones.
- **Siguiente/Previo objeto animado.** 

Estos comandos nos servirán para desplazarnos en un objeto agrupado, haciéndonos más fácil configurar las animaciones; no será necesario desagrupar el objeto para acceder a los elementos que se encuentran debajo de otros.
- **Habilitar variables de escritura.** 

Será imprescindible marcar esta opción de la barra de herramientas durante la simulación, sino está activada no podremos modificar el valor de ninguna variable mediante los botones u otros elementos que tengamos en la pantalla de operador.

4.3 Ejemplos de elementos creados.

Explicadas las herramientas usadas se van a poner ejemplos prácticos y cómo se han ido utilizando para generar las partes de la pantalla de operador. Además se incidirá en las animaciones que se han usado explicando las más relevantes.

4.3.1 Cuadro de mandos.

En esta zona del gráfico de la instalación se encuentran todos los botones con los que cuenta la instalación.



Figura 31. Gráfico cuadro de mandos.

Los pasos para crearlo han sido los siguientes:

1. Se crean las zonas geométricas como son el recuadro exterior gris y los interiores: el recuadro amarillo, el verde, el rojo. Configurando sus propiedades para que tengan los colores nombrados y jugando con la opción de primer plano y segundo para que adquiera el aspecto que se desea.
2. Se colocan los botones. Reset, Continuar y Vaciar son del tipo botón de comando; parada de emergencia es de casilla de verificación y Stop/Run es un cuadro de giro.
3. Se asignan las variables y el nombre que muestran. En botones de tipo comando podremos elegir entre que actúe como un pulsador ('1' lógico sólo mientras está pulsado) o bien que sea un botón con *retención* (aguenta el valor hasta que se vuelva a pulsar). Todo ello se ilustra en la Figura 32:

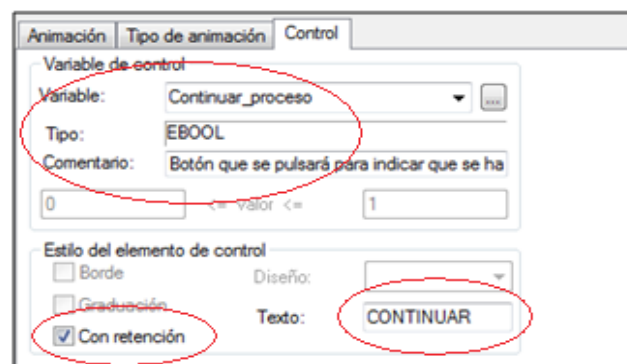


Figura 32. Configuración de botón de comando.

4. Desde la *librería de pantallas de operador*, dentro de *accionadores*, se inserta el botón de la *Figura 31*. Una vez insertado nos valemos de los botones de *Siguiente/Previo objeto animado* para seleccionar las flechitas amarillas y las animamos como se indica en la *Figura 33*:

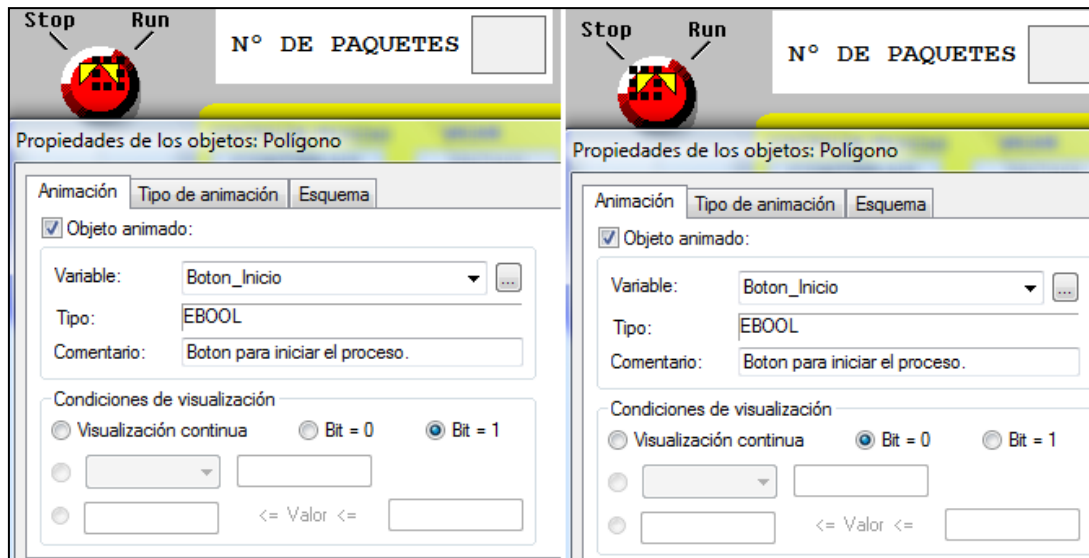


Figura 33. Animación de botones.

5. Insertar un *campo de entrada* para que nos vaya mostrando el valor de la variable que cuenta el número de paquetes, se ha de configurar como se observa en la *Figura 34*:

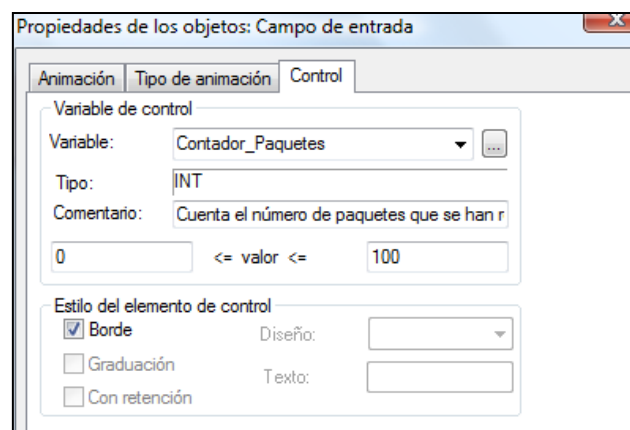


Figura 34. Configuración de un campo de entrada.

6. Por último se han de introducir los últimos letreros que falten ayudándonos del botón *insertar cuadro de texto*, le daremos la apariencia necesaria y el tamaño que se desea. Tras todo ello seleccionamos todos los objetos y los agrupamos para que nos quede como un único elemento y sea más sencillo desplazarlo a la zona de la pantalla de operador que se desee.

4.3.2 Simulación de sensores.

En la *Figura 31*, en la que se muestra el gráfico completo de la instalación, podemos ver hasta cuatro sensores de posición, dos sensores de nivel más en el depósito y un sensor más de nivel para realizar el control de calidad. Dado que en el modo simulación no podemos asociar direcciones a variables, para simular estas variables podemos optar o por una tabla de animación (también incluidas en el proyecto), o bien para ir modificando el proceso en la propia pantalla se han incluido botones que nos ayudarán a variar el valor de dichas variables. Además se ha dispuesto de un elemento que nos indique si están a '1' o '0'. Desarrollaremos un caso pero todas ellas se configuran de la misma forma.

Desde la *librería de pantallas de operador*, dentro de *accionadores*, se inserta el indicador que tenemos el gráfico y que se muestra en la *Figura 35* agrupado y sin agrupar.

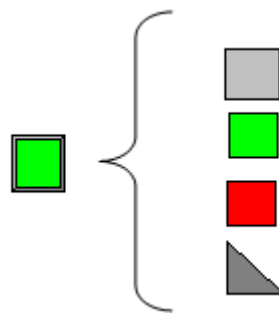


Figura 35. Elemento agrupado y desagrupado.

La parte roja desagrupada de corresponderá con el sensor desactivado ('0') y la parte verde hace referencia al sensor activado ('1'); cada uno de la variable que corresponda, configurados de igual forma que se hizo en la *Figura 33*.

Para controlar el valor de cada variable asociada a estos elementos animados, como se ha comentado, se asociarán *botones de comando* a cada uno para ir simulando el proceso; en el caso del sensor situado en la zona de control de calidad será una *casilla de verificación*, configurada de igual forma que el anterior elemento.

4.3.3 Simulación de salidas.

En la pantalla gráfica también aparecen varios elementos que simulan actuadores y que se activan con las respectivas salidas del proyecto. Dentro de los actuadores se tienen algunos que se han creado con formas sencillas con las herramientas gráficas que proporciona el programa (*empaquetadora* y *etiquetadora*); otras son válvulas sacadas de la librería (*V0*, *V1*, *V_deasgue*), su manera de configurarla es exactamente igual a las de los sensores o lo explicado en la *Figura 33*, usando los botones de *siguiente/previo animado*. Un caso especial es de la taponadora ya que se ha usado un elemento de la biblioteca y adaptado a este proyecto, se desagrupó y eliminó las partes que no eran útiles.

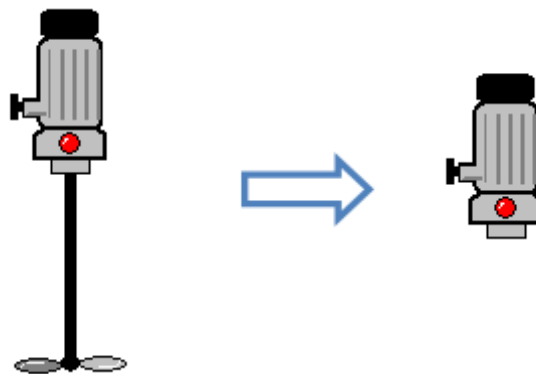


Figura 36. Creación de la taponadora.

Partiendo de una hélice para mezclar fluidos, se desagrupa el objeto y se elimina las partes innecesarias para obtener una imagen que puede hacer las veces de una taponadora. Además el interruptor que lleva se configura de la forma que se ponga a verde cuando se activa la salida.

Otras salidas son las cintas transportadoras, las imágenes permanecen estáticas, pero lo que sí está animado son los interruptores de estado. Configurado de igual forma que los sensores y añadidos un cuadro de texto que se muestran uno u otro según el valor de cada variable. Además cuando las cintas estén activadas aparecerán unas flechas para dar la idea de movimiento.

4.3.4 Simulación del nivel del depósito.

Una parte especial de la simulación es el depósito, este elemento lleva asociado una variable tipo INT y no EBOOL como el resto de elementos explicados hasta ahora. Las etapas a seguir son las siguientes:

1. En primer lugar buscamos el depósito dentro de los elementos de la librería.
2. Tras ello usando las herramientas explicadas para movernos dentro de los objetos agrupados, se selecciona el rectángulo central y accedemos a sus propiedades.
3. En la pestaña *Tipo de animación*, marcamos la opción *gráfico de barras* y accedemos a seleccionar el tipo de gráfico de barras que se desea, los límites y el color que se mostrará; para más detalle consultar la *Figura 37*.

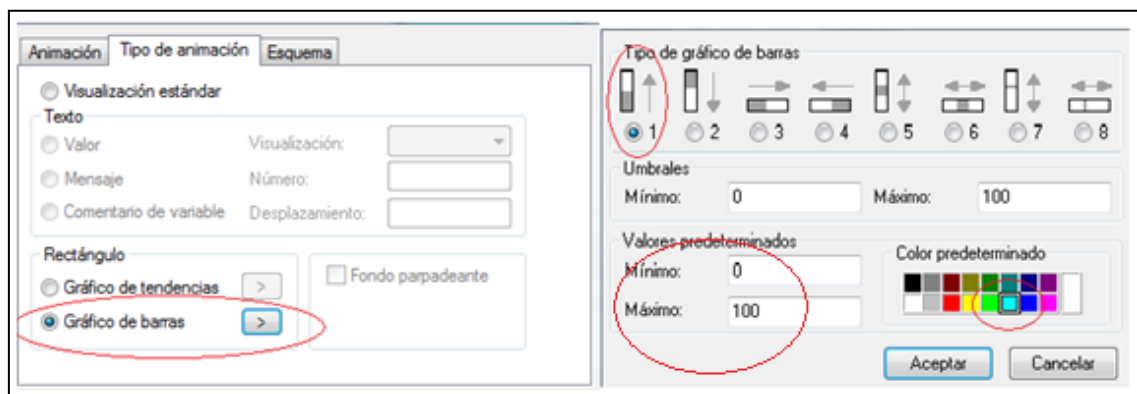


Figura 37. Configuración de un gráfico de barras.

Con esta configuración podremos ver gráficamente como el nivel del tanque va variando según varíe el valor de la variable asociada a ello (*nivel_deposito*).

4.3.5 Simulación del movimiento de botellas.

En este caso, al igual que en el nivel del depósito del tanque, la variable que hace posible esta animación es un entero (*Posicion*), ahora no lleva asociado un gráfico de barras, sino que en la programación este entero va tomado distintos valores que son los que se asocian a las diferentes imágenes precargadas (por medio de la opción *insertar imagen*); de tal forma que según en la parte de proceso que estemos la variable toma un valor y aparece la imagen adecuada. Para configurar la imagen se ha de seguir la *Figura 38*.

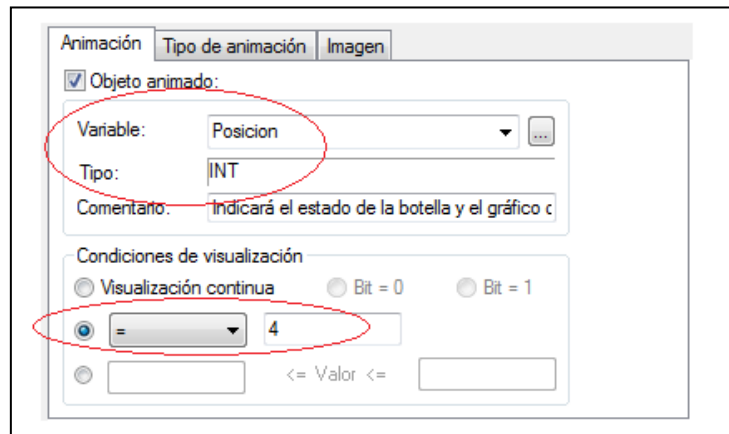


Figura 38. Configuración simulación de botellas.

La secuencia gráfica que sigue la botella a lo largo del proceso se muestra en la *Figura 39*, se ha de tener en cuenta que dependiendo de si la botella es declarada apta o no por el control de calidad, adquirirá un aspecto u otro.

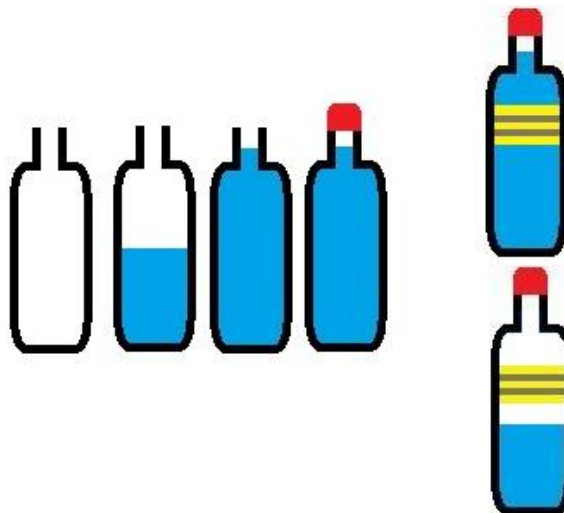



Figura 39. Evolución gráfica de la botella.

CAPÍTULO V – Evolución del sistema.

A continuación se va a ver la evolución gráfica del comportamiento del sistema, según se vayan produciendo las transiciones necesarias. **Es muy importante pulsar el botón que permite habilitar las variables de escritura**, de lo contrario los botones estarían inutilizados. Tras ello seleccionamos el *modo simulación* en el menú *PLC* y en este mismo menú pulsamos *conectar* y *transferir proyecto al PLC*. El sistema estará listo sólo falta dar comienzo al proceso

con accionando el botón **RUN**  y la instalación tomará el siguiente aspecto:

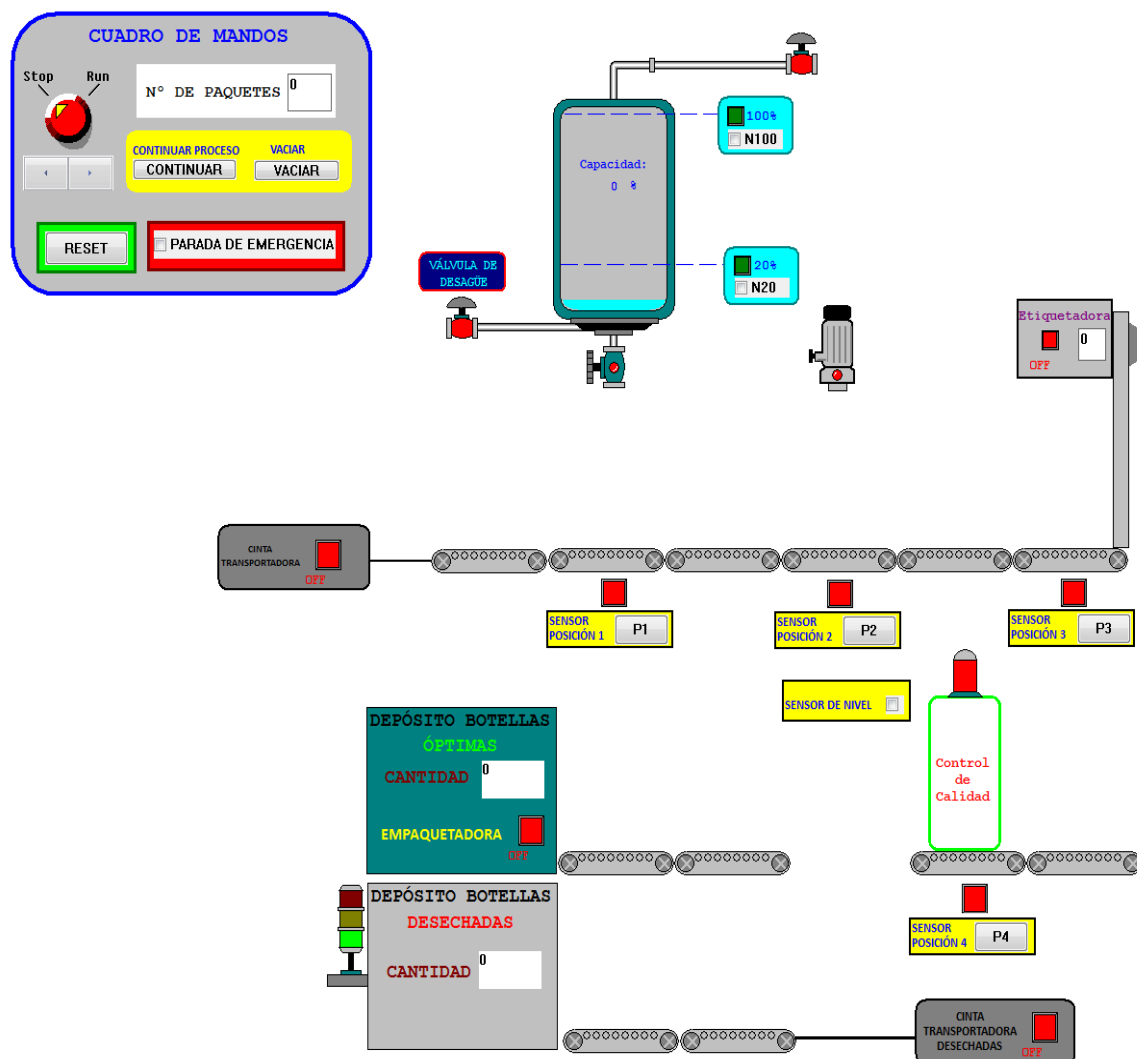
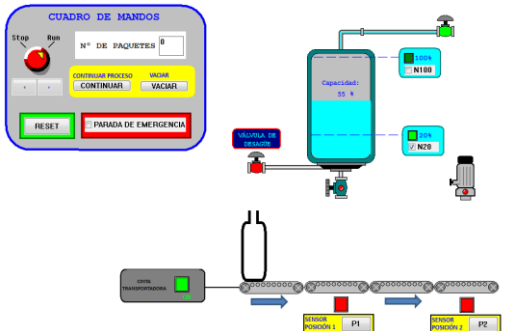
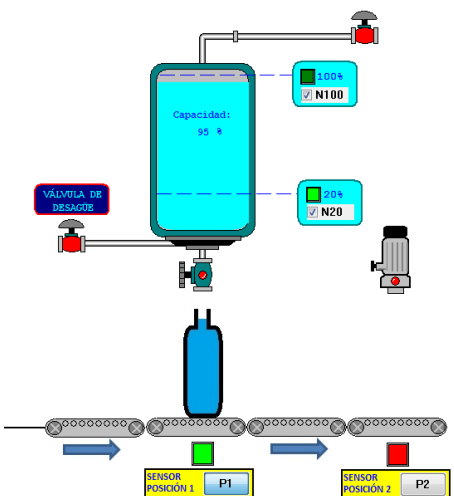
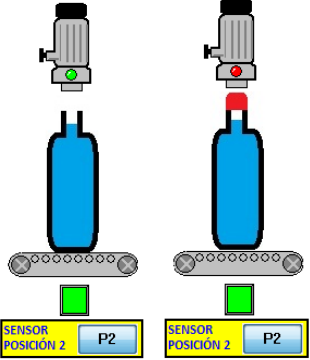
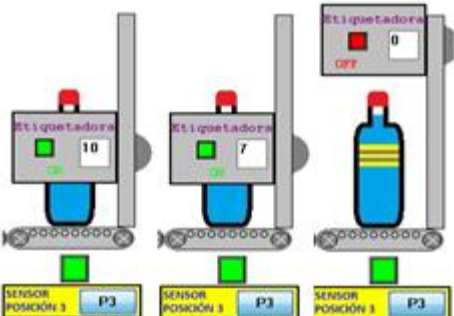


Figura 40. Aspecto de la instalación en reposo.

DESCRIPCIÓN	VISUALIZACIÓN GRÁFICA
<p>I) Tras pulsar en la pantalla de operador el botón de <i>RUN</i> dará comienzo el proceso, el tanque se empezará a llenar y la primera botella aparecerá cuando el depósito alcance un mínimo del 20%, cuando lo alcance deberemos marcar <i>N20</i> para indicarlo. En el momento que aparece la botella se activa el motor de la cinta. Par detener el llenado del tanque deberemos activar <i>N100</i>.</p>	 <p>Figura 41. Inicio del proceso.</p>
<p>II) Una vez que se pulsa el botón <i>P1</i> en la pantalla de operador la botella comienza a llenarse, se muestra medio llena y llena del todo, la capacidad del tanque decrece en un 5% y comienza a moverse la cinta.</p>	 <p>Figura 42. Botella llena.</p>
<p>III) Debemos desactivar <i>P1</i>, de lo contrario la botella no continuará su movimiento. Hará un paso a la mitad de la cinta y tras pulsar <i>P2</i> comenzará el taponado de la botella.</p>	 <p>Figura 43. Proceso de taponado.</p>
<p>IV) Tras desactivar <i>P2</i> y pulsar <i>P3</i>, llegará a la zona de etiquetado. Donde primero baja la etiquetadora y comienza el proceso de etiquetado que dura diez segundos.</p>	 <p>Figura 44. Secuencia de etiquetado.</p>

V) Una vez desactivado el sensor previo y activado *P4* se llega a la zona de control de calidad, donde permanece unos segundos. Aquí la botella seguirá dos posibles caminos: si marcamos *sensor de nivel* la botella irá hacia la zona de empaquetado; o si por el contrario no se marca, hacia la zona de desecho.

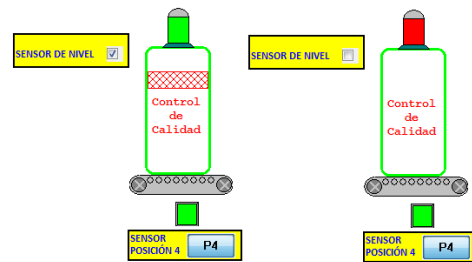


Figura 46. Opciones control calidad.

VI.A) En el caso de que la botella haya sido declarada como válida se desplazará hasta la zona de empaquetado.

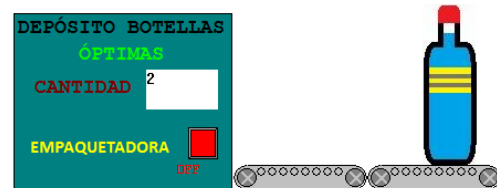


Figura 47. Botella en zona de empaquetado.

VII.A) Si llegan tres botellas a la zona de empaquetado se activará la empaquetadora. Reiniciándose el número de botellas en esta zona y aumentando en una unidad el número de paquetes realizados.



Figura 48. Activación de la empaquetadora.

VI.B) Si la botella no ha pasado el control de calidad será llevada a una zona de desechos.

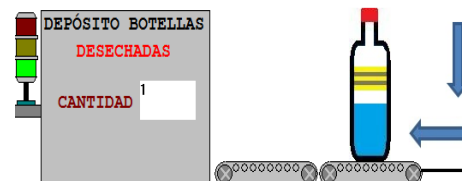


Figura 49. Botella en zona de desecho.

VII.B) Si en la zona de desechos se alcanzan dos botellas se encenderá el indicador amarillo, pero con tres de ellas se encenderá la luminaria roja, se activará la alarma y se interrumpirá el proceso, a la espera de que un operario pulse el botón *Continuar*, que indicará que todo está correcto y se han retirado las botellas descartadas.

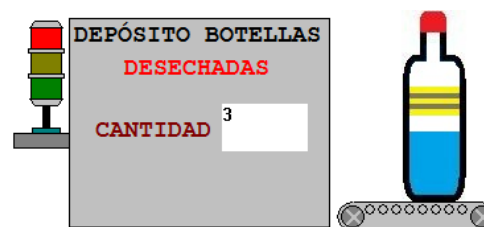


Figura 50. Límite de botellas desechadas.


<p>VIII) Si en algún momento se pulsa el botón de <i>Vaciar</i>, se terminará la botella que se está procesando y tras lo cual se detendrá el proceso y comenzará a vaciarse el depósito.</p>	 <p>Figura 51. Vaciado del depósito.</p>
<p>IX) Si en algún momento de la ejecución pulsamos el botón de <i>Reset</i> se reiniciarán los contadores. Si pulsamos <i>Parada de emergencia</i> saltará la alarma y se detendrá el proceso, tras desactivarse dicha entrada el proceso continuará justo por donde se había quedado.</p>	 <p>Figura 52. Elemento de alarma.</p>

TABLA 8. Evolución de la simulación.

CAPÍTULO VI – Modo Docente.

Una vez creado el sistema con una solución acorde a lo planteado, diseñada la pantalla de operador con todos sus elementos y teniendo muy clara la evolución del proceso, se ha dado un último paso. El origen de este trabajo fin de grado está en la aplicación docente que ofrecen el simulador y la pantalla de operador de Unity Pro y en este apartado se aborda cómo se pretenden usar ambas capacidades conjuntas.

La esencia de este modo es que el alumno no tiene la solución a su alcance, es el propio alumno el que se encarga de configurar y programar el sistema para obtener los objetivos especificados en el enunciado.

Para ello, como es de suponer, se han eliminado todos los diagramas SFC y las hojas de contactos creadas; por lo que ha desaparecido toda programación previa incluida la de la pantalla de operador. En este caso nos interesa crear una programación que sólo se encargue del control de la pantalla de operador. Se pretende que si el alumno está llenando la botella pero no activa la válvula de llenado, gráficamente vea que la botella no se llena; o bien, otro ejemplo podría ser que al llenar dicha botella se dejara abierta la válvula del tanque, en tal caso se vería como el propio tanque se va vaciando. Así el alumno podrá reaccionar y revisar aquellas etapas que esté viendo que el resultado no es el esperado.

La única restricción encontrada a este modo es el nombre de las variables, se le debe proporcionar al alumno una lista (definida en el propio programa) con el nombre de las variables y el tipo que son. El alumno debe usar estos nombres, de lo contrario la animación no ofrecerá cambio alguno.

Como se ha comentado se han eliminado las cuatro secciones anteriores que se encargaban de hacer funcionar del sistema, a cambio, se han programado dos nuevas secciones una en lenguaje ST y otra en LD. En este punto vemos de nuevo la versatilidad de Unity Pro, dado que ambos procesos están relacionados entre sí y se pasan información de variables. Describiremos, a continuación, las funciones que adquiere cada una de estas secciones:

- Sección “Condiciones”.

Programada en lenguaje ST. Se encarga de definir todas las condiciones necesarias para configurar gráficamente la posición de la botella. Cómo se ha explicado sólo nos centramos en las condiciones de las entradas y las salidas, comentaremos un ejemplo para ilustrar este módulo, mostrado en la *Figura 53*. En este trozo de código vemos que es independiente de donde provengamos, si se activa el sensor *Posicion1* y el resto están desactivados vamos a colocar la botella justo debajo del tanque. El tiempo lo mide la variable *Auxiliar* en segundos, mientras que sea menor que tres segundos la botella se mantendrá vacía, de tres segundos a cinco medio llena y a los ocho segundos totalmente llena. Que la botella se muestre medio llena y llena no sólo depende del tiempo, sino de que el alumno haya activado la válvula *V1* encargada de verter su contenido. Se aprecia también una variable *Pcinta*, dicha variable nos permitirá apreciar el movimiento de la botella en la cinta, cuando se desactive el sensor *Poscion1* y se active *M_cinta*.

```

(* Tratamos el llenado de la botella *)
IF (Posicion1 = TRUE) AND (Posicion2 = FALSE)
AND (Posicion3 = FALSE) AND (Posicion4 = FALSE) THEN

    Pcinta:=1;

    if (Auxiliar<3) THEN
        Posicion := 1;

    elsif (Auxiliar>=3 AND Auxiliar<5 AND (V1 = TRUE) ) THEN
        Posicion := 2;

    elsif (Auxiliar>=5 AND (V1 = TRUE) ) THEN
        Posicion := 3;

    END_IF;
END_IF;

```

Figura 53. Código sección condición, modo docente.

- Sección “Tiempo y deposito”.

Programada en hoja de contactos (LD). Tiene dos propósitos, por un lado medir el tiempo que está la botella en cada zona de embotellado y por otro controlar el vaciado y llenado del tanque.

Para controlar los tiempos se ha creado una variable *Auxiliar*, esta variable es de tipo entero (INT) y se va incrementando una unidad cada vez que %S6 emite un pulso positivo, es decir, cada segundo. Esta variable se reiniciará cada vez que haya un flanco positivo o negativo de los sensores de posición. El análisis de los tiempos se hace en la sección programada en ST.

Una parte especial del control de tiempos es controlar el tiempo del campo de entrada de la etiquetadora, para ello basta con hacer una simple resta, dentro de los tiempos indicados y con el sensor de posición adecuado, como se muestra en la *Figura 54*:

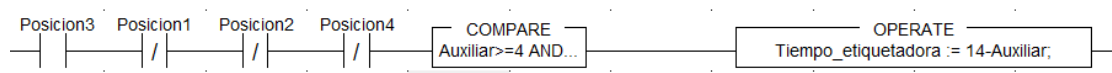


Figura 54. Control del display etiquetadora, modo docente.

Dentro del control del nivel del depósito también se ha incluido la bajada del depósito durante el llenado de cada botella. La lógica empleada es simple: si tengo la válvula *V1* abierta estoy vertiendo líquido a la botella. Como el nivel del tanque decrece un 5% por cada botella y tardo 5 segundos, mientras esté abierta *V1*, *Nivel_deposito* decrecerá en una unidad. Si el alumno deja esta salida activada más tiempo de lo debido el tanque se vaciará progresivamente.

La programación completa y detallada de los dos módulos se incluye en el *Anexo I – Programación Modo Docente*.

CAPÍTULO VII – Comparativa Modo simulación - PLC.

7.1 Descripción del PLC usado.

EL autómatas en el cual se ha probado el proyecto realizado, ha sido uno de los presentes en los laboratorios de automatización, donde realizan sus prácticas los alumnos que cursan la asignatura de Automatización Industrial I. Dentro de las tres familias de PLC's que posee Schneider Electric, la presente en los laboratorios es la gama *Premium*, más concretamente el modelo *TSX P57 104M 0260 57-1*. El autómatas está compuesto por los siguientes elementos: ^[6]

- Bastidor.
 - Procesador.
 - Fuente de alimentación: *TSX PSY 2600M*.
 - Módulo de 16 entradas digitales: *TSX DEY 16FK*.
 - Módulo de 16 salidas digitales: *TSX DSY 16R5*.
 - Módulo de entradas analógicas. *TSX AEY 414*.
 - Módulo de salidas analógicas. *TSX ASY 410*.
- Telefast.
- Cable comunicación con PC.



Figura 55. Autómatas del laboratorio de Automatización de UC3M.

Toda esta configuración se llevará al proyecto de Unity Pro, para conseguir una adecuada comunicación al transferir el proyecto al autómatas.

7.2 Transferencia del proyecto.

Una vez realizado el proyecto entero deberemos compilarlo y generarlos para estar seguros de que no existe ningún tipo de error, aunque estas comprobaciones ya se han ido haciendo poco a poco para que los errores fueran más fáciles de subsanar.

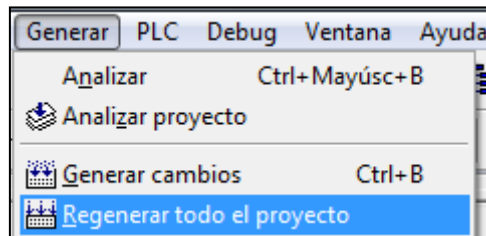


Figura 56. Pestaña de generar proyecto.

Tras generar nuestro proyecto y modificados los posibles errores, deberemos establecer la dirección adecuada, tanto si queremos usar el *modo simulación* o el *modo estándar* utilizando el PLC. Para ello debemos configurar los parámetros pulsando en el menú *PLC* y seleccionando *Establecer dirección...*; tal y como se muestra en la Figura 55:

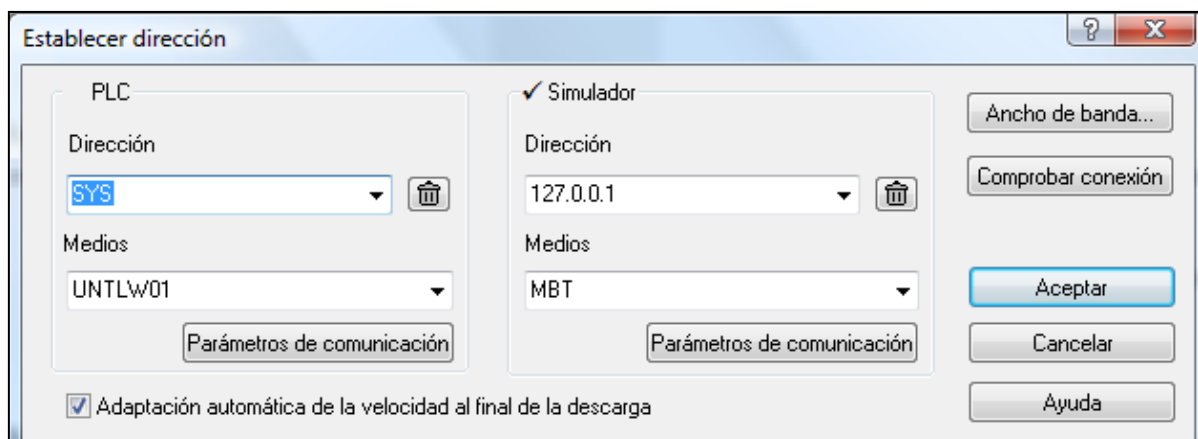


Figura 57. Configuración de dirección.

Ajustados todos los parámetros ya podremos elegir la *modalidad estándar* (PLC) o bien la *modalidad simulación* del menú *PLC*. Tras lo cual conectaremos con el autómata o el simulador (según hayamos escogido) y transferiremos el proyecto. Todo este desarrollo se muestra con detalle en la Figura 56.

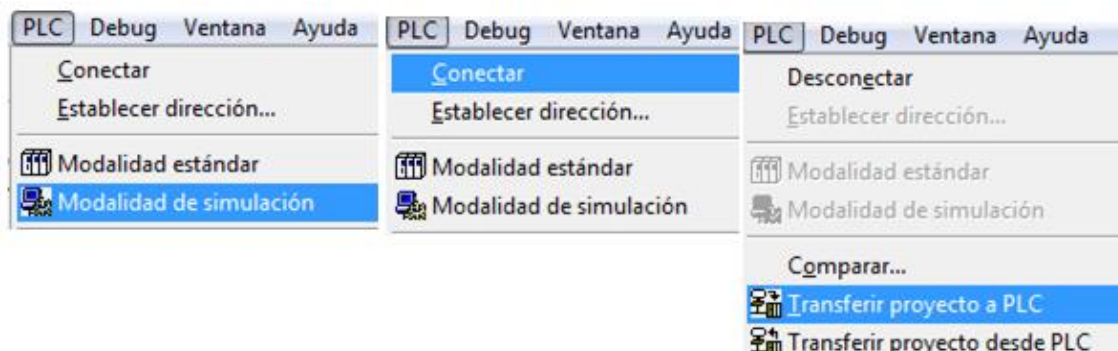


Figura 58. Proceso para transferir un proyecto.

7.3 Comparativa Simulador – PLC.

Como siempre que tratamos con un simulador, van a existir diferencias con respecto al comportamiento que habría con la máquina en cuestión, en este caso con nuestro autómatas. Algunas desventajas detectadas del simulador respecto a un PLC convencional son: ^[9]

- El simulador de PLC no admite ninguna forma de E/S. Aunque la simulación contiene los componentes del proyecto para E/S, el simulador de PLC no los procesa. A las entradas y salidas sólo se puede acceder desde el proyecto o a través de las funciones online de Unity Pro (leer, escribir, forzar, animar, etc.).
- A pesar de que el simulador es capaz de incluir en su simulación el proceso completo, con todas las tareas de usuario, la velocidad de proceso del simulador durante la ejecución es sensiblemente inferior a la que podemos encontrar en un PLC real.
- No todos los bits de sistema están disponibles mientras estamos simulando, entre ellos el bit de rearranque en frío %S0 es uno de ellos.
- El simulador del PLC no soporta ninguna característica de *Hot Standby*.
- El simulador del PLC sólo admite la comunicación basada en TCP/IP (puerto 502 de Schneider). En los demás casos se emitirá un mensaje de error de Modbus.
- El simulador del PLC no tiene una ejecución de doble código de la lógica y la comparación de los resultados. Se limita a simular la lógica del PLC, pero no simula su comportamiento de seguridad.

A pesar de presentar estas desventajas, el simulador no deja de ser una gran arma didáctica, que al fin y al cabo, es el fin que en este caso se requiere. El simulador será accesible a cualquier alumno con sólo tener el software en un ordenador, lo que solucionará los problemas de accesibilidad a los puestos con los autómatas y será una solución mucho más económica y sencilla que el PLC. Antes los alumnos sólo podía realizar sus pruebas en los PLC del laboratorio, pero ahora al ampliarse notablemente las posibilidades, el alumno podrá dedicarle más tiempo afianzar sus conocimientos y permitirá al personal docente nuevas formas de trabajo.

7.4 Diferencias entre el simulador y un sistema SCADA.

Un sistema SCADA (Supervisory Control And Data Acquisition – Supervisión Control y adquisición de Datos) es un software para ordenadores que permite controlar y supervisar procesos industriales a distancia. Facilita retroalimentación en tiempo real con los dispositivos de campo (sensores y actuadores) y controlando el proceso automáticamente. Provee de toda la información que se genera en el proceso productivo y permite su gestión e intervención.^[7]

La mayor diferencia entre ambos radica en que el sistema SCADA es un sistema de supervisión, mientras que un simulador, como su propio nombre indica, se encarga de simular las variables pudiendo incidir en ellas. SCADA supervisa las tareas de uno o varios autómatas, pudiendo o no tener control sobre ellos, dependiendo de cómo sea la relación y la jerarquía entre ellos; pero casi siempre son labores de supervisión. En muchos casos el SCADA tiene retroalimentación o *feedback*, que corresponde al proceso de recabar información para mejorar o modificar diversos aspectos de funcionamiento de una organización, para que la mejora sea continua esta retroalimentación debe ser bidireccional.

Unity Pro no es un sistema SCADA, la propia marca Schneider tiene su software dedicado a estas tareas, Unity Pro se puede configurar en un sistema SCADA con la conveniente conexión vía Ethernet y la asignación correspondiente de puertos.

CAPÍTULO VIII – Presupuesto.

Código	Unidad de Medida	Descripción	Medición	Precio Unitario	Precio Total
01	CAPÍTULO 1: Hardware				
01.01	ud.	<u>Autómata TSX P57 104M 0260 57-1.</u> Autómata de la marca Schneider de la gama Premium, que incluye un procesador capaz de gestionar toda la estación PLC.	1	823,20 €	823,20 €
01.02	ud.	<u>Fuente de alimentación: TSX PSY 2600M.</u> Es un módulo de alimentación de formato simple de corriente alterna. Tensión nominal: 100 - 240V. Tensión límite: 85...264V. Potencia aparente 50 VA. Frecuencia nominal; 50-60 Hz.	1	714,40 €	714,40 €
01.03	ud.	<u>Módulo de 16 entradas digitales: TSX DEY 16FK.</u> Permiten configurar el tiempo de filtrado de las entradas entre 0 y 7,5 ms (4 ms por defecto). Se pueden configurar hasta 16 entradas que permiten registrar los sucesos y su tratamiento inmediato por parte del procesador (tratamiento al producirse interrupción).	1	609,60 €	609,60 €
01.04	ud.	<u>Módulo de 16 salidas digitales: TSX DSY 16R5K.</u> Tensión de funcionamiento 24Vcc, corriente 3A. Lógica positivo/negativa. Sin protección.	1	561,60 €	561,60 €
01.05	ud.	<u>Módulo de entradas analógicas. TSX AEY 414.</u> Propone 23 rangos de cada una de sus entradas que es posible configurar canal por canal.	1	1.319,20 €	1.319,20 €
01.06	ud.	<u>Módulo de salidas analógicas. TSX ASY 410.</u> El módulo TSX ASY 410 es un módulo de 4 salidas aisladas entre sí. Para cada una de ellas, ofrece las gamas: tensión +/- 10 V, corriente 0..20 mA y 4.. 20 mA.	1	1.650,40 €	1.650,40 €
01.07	ud.	<u>Telefast.</u> Sistema precableado que permite la conexión y adaptación de señales de control de tarjetas PLC equipadas.	1	125,63 €	125,63 €
01.08	ud.	<u>Cable de comunicación con el PC.</u> Cable utilizado para transminitr la información entre el PLC y el ordenador del usuario.	1	32,00 €	32,00 €
TOTAL CAPÍTULO 1: HARDWARE					4.516,86 €

02 CAPÍTULO 2: Software

02.01	ud.	<u>Licencia Unity Pro.</u> Autorización del uso del programa de Schneider para aplicaciones docentes durante el plazo de dos años.	1	1.600 €	1.600 €
TOTAL CAPÍTULO 2: SOFTWARE					1.600 €

En el presupuesto presentado se han incluido todos los componentes necesarios para llevar a cabo la programación en un PLC, se desarrollan dos capítulos diferenciados entre la parte de software y de hardware. Además sería necesario disponer un ordenador en el que tener instalado el software de *Unity Pro*. Un resumen, por capítulos, del presupuesto presentado es:

RESUMEN DEL PRESUPUESTO	
CAPÍTULO 1: HARDWARE.	4516,83 €
CAPÍTULO 2: SFOTWARE.	1600 €
<u>TOTAL PRESUPUESTO</u>	6.116,83 €

TABLA 9. Resumen presupuesto.

CAPÍTULO IX – Conclusiones y aplicaciones futuras.

9.1 Conclusiones.

Al igual que en otras áreas, la automática se abre a los nuevos tiempos y comienzan a aparecer los primeros simuladores, como el que presenta Unity Pro. Como todo en sus comienzos se encuentran defectos y cosas por mejorar pero hay que reconocer que la herramienta que introduce Schneider en este software, unido con las pantallas de operador, ofrece un gran abanico de posibilidades y tiene un gran potencial.

Dentro de las cosas que se han de mejorar en lo referente al simulador son: adaptarlo para que pueda hacer las mismas funciones que cualquier autómatas, mejorar su velocidad de procesamiento para asimilarla a la del autómatas, admitir E/S y mejorar sus prestaciones en lo referente a seguridad y prestaciones *Hot Standby*.

La interfaz gráfica o pantalla de operador es un complemento ideal al simulador, sobre todo en el ámbito didáctico. Es útil tanto al profesor, a la hora de comentar los conceptos y su puesta en práctica; como para el alumno, facilitando su práctica y dinamizando su aprendizaje. Parte muy importante en ese aprendizaje es que la pantalla del programa en sí y la pantalla de operador en concreto, son especialmente sencillas de utilizar. Uno de los puntos más débiles es la poca variedad de recursos gráficos de la librería, llegando en muchos casos a parecer dibujos animados; de ahí que sea uno de los principales aspectos a mejorar, para poder lograr representar entornos de una manera más seria y profesional.

En este documento se ha desarrollado con detalle un problema propuesto, analizando paso a paso la forma de resolverlo, para que sirva como guía práctica de manejo del programa, siempre desde el punto de vista didáctico. También se ha demostrado la versatilidad de Unity Pro, en el que la estructura principal del programa se hace en SFC (GRAFCET), pero que la programación a nivel más bajo se realiza en LD y ST. Sobre todo su versatilidad se ha notado a la hora de resolver el problema de la parada de emergencia, al no poder usar eventos, se ha hallado la forma combinando otros recursos para obtener el mismo resultado.

Ante la imposibilidad de llevar a cabo la implementación de la parada de emergencia mediante un evento, se contactó con el departamento técnico de Schneider Electric, con el objetivo de conocer qué se estaba realizando de manera errónea, dado que el programa no accedía a los eventos. Se les explicó la situación y el propósito de realizar el evento, tras varias consultas los empleados de Schneider Electric comunicaron que el proceso llevado a cabo era correcto, pero que los eventos de E/S sólo se podían activar mediante determinados tipos de tarjetas de contaje; de lo contrario el autómatas no estaría en disposición de realizar dichas tareas. Esta información no estaba disponible en ninguno de los manuales de Unity Pro. Debido a que los autómatas de los laboratorios no disponían de dicha tarjeta, se optó por aportar otra solución sin estar basada en eventos.

En conclusión, Unity Pro ofrece todas las posibilidades que se pueden demandar didácticamente hablando. A pesar de las restricciones que actualmente ofrece el programa, éstas sólo afectarían a aplicaciones en un ámbito profesional. Para el aprendizaje de los alumnos será de gran utilidad, no serán necesarios los autómatas para comprobar los diseños, al tener el simulador en cualquier ordenador; todo ello hará la asignatura mucho más dinámica e interesante para el alumno.

9.2 Aplicaciones futuras.

En lo referente a las aplicaciones futuras, podemos distinguir dos vertientes: aplicaciones para el programa creado en este proyecto y otra el software desarrollado por Schneider Electric.

Este trabajo puede desarrollarse y ampliarse, por un lado se puede crear otra pantalla de operador que trate la etapa previa que recibe unas botellas sucias y las deja listas para usarse en el proceso que se ha implementado. Así se dispondrían de dos pantallas de operador funcionando a la vez.

También se podría implementar la parada de emergencia del modo que se había pensado en un principio, mediante un evento de E/S. Para ello sería necesario que los autómatas contaran con la tarjeta TSXDMY28FK, así sería posible activar los eventos. Si se dispusiera de ello sería interesante implementar tanto un Watchdog que vigilase que el sistema funciona correctamente en todo momento; así como controlar entradas analógicas con otro Watchdog. Con esta clase de tarjetas se podría llevar a cabo una programación más avanzada que puedan ser impartidas en otras asignaturas de más alto nivel que *Automatización Industrial I*. Aún así, sería posible configurar el programa con eventos, siempre y cuando añadiéramos la tarjeta indicada en el autómata del programa y todo ello podría ser simulado. Con la simulación podemos ver todas las posibilidades que nos ofrecen estas y otras variantes, para después decidir si adquirir dicha tarjeta. Esto abre muchas posibilidades en el ámbito profesional para facilitar el éxito de futuras modificaciones, al tener la información de cómo se comportan en la simulación.

Respecto al programa Unity Pro, se adapta perfectamente a las necesidades de los autómatas del laboratorio y a los conceptos impartidos en la asignatura de *Automatización Industrial I*. El simulador aumenta en número de PLC's disponibles a un coste mucho menor, el precio de un autómata completo ronda los 4500€ por los 1600€ de la licencia que podremos instalar en cualquier ordenador, todo ello conlleva a un aumento de recursos con un gasto comedido.

Para una aplicación profesional, Unity Pro ofrece muchas más limitaciones. En el ámbito de programación del PLC la competencia es numerosa y marcas como *Siemens* ofrecen un software de alto nivel y altas prestaciones. El simulador es un buen comienzo pero para ser útil en un entorno profesional y poder usarse para ver evoluciones de sistemas con gran precisión, se requieren que se asemeje totalmente al funcionamiento de un autómata; así se podrían ver cómo influirían posibles cambios en la planta, ayudar a eliminar cuellos de botella... Además de introducir mejoras en la pantalla de operador que le den un aspecto más serio y profesional.

Por último, destacar la labor encomiable de Schneider y dar el primer paso al crear este simulador que, a pesar de sus limitaciones, es capaz de recrear aplicaciones sencillas de la misma forma que lo haría un PLC.

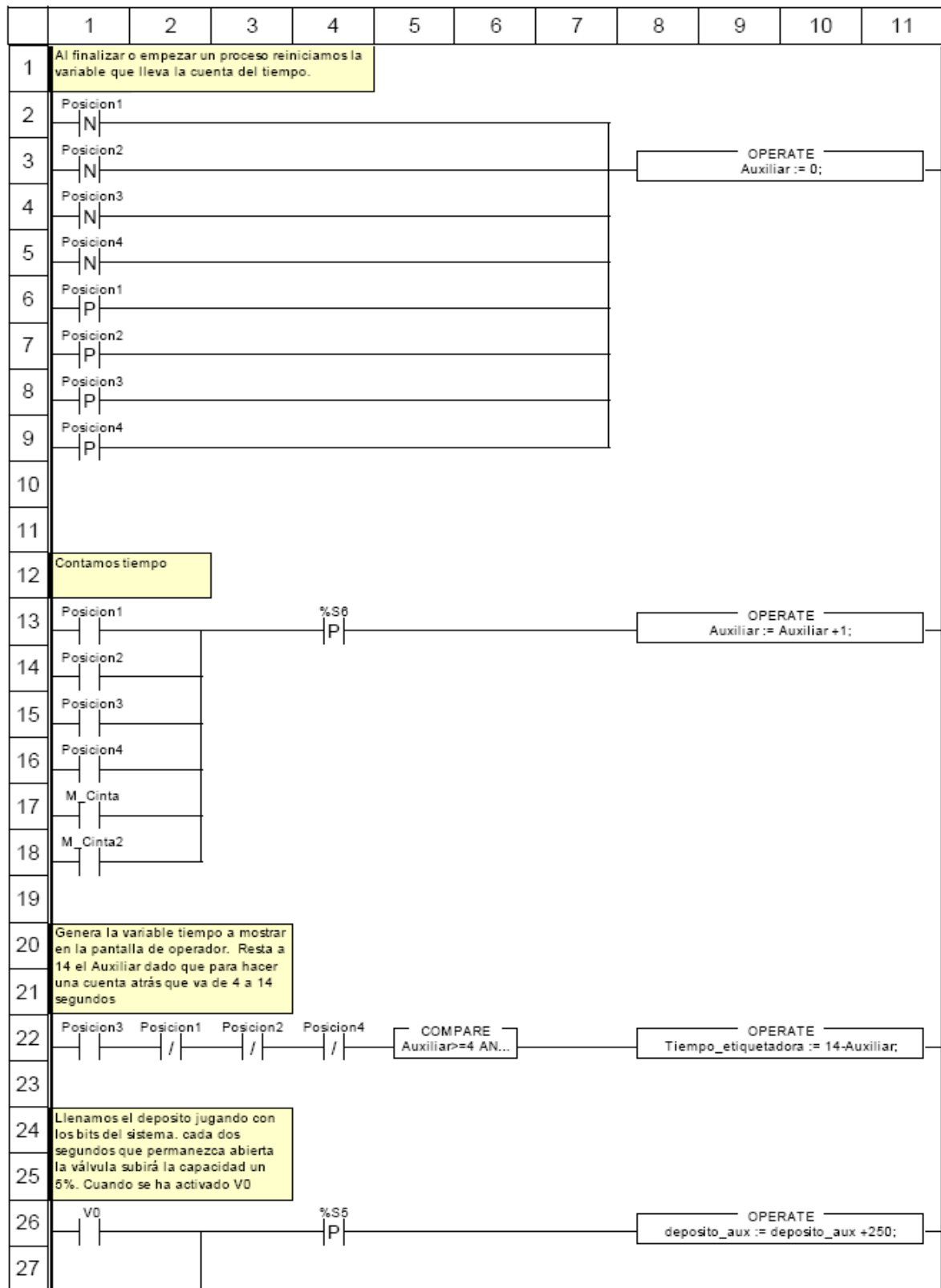
10. BIBLIOGRAFÍA.

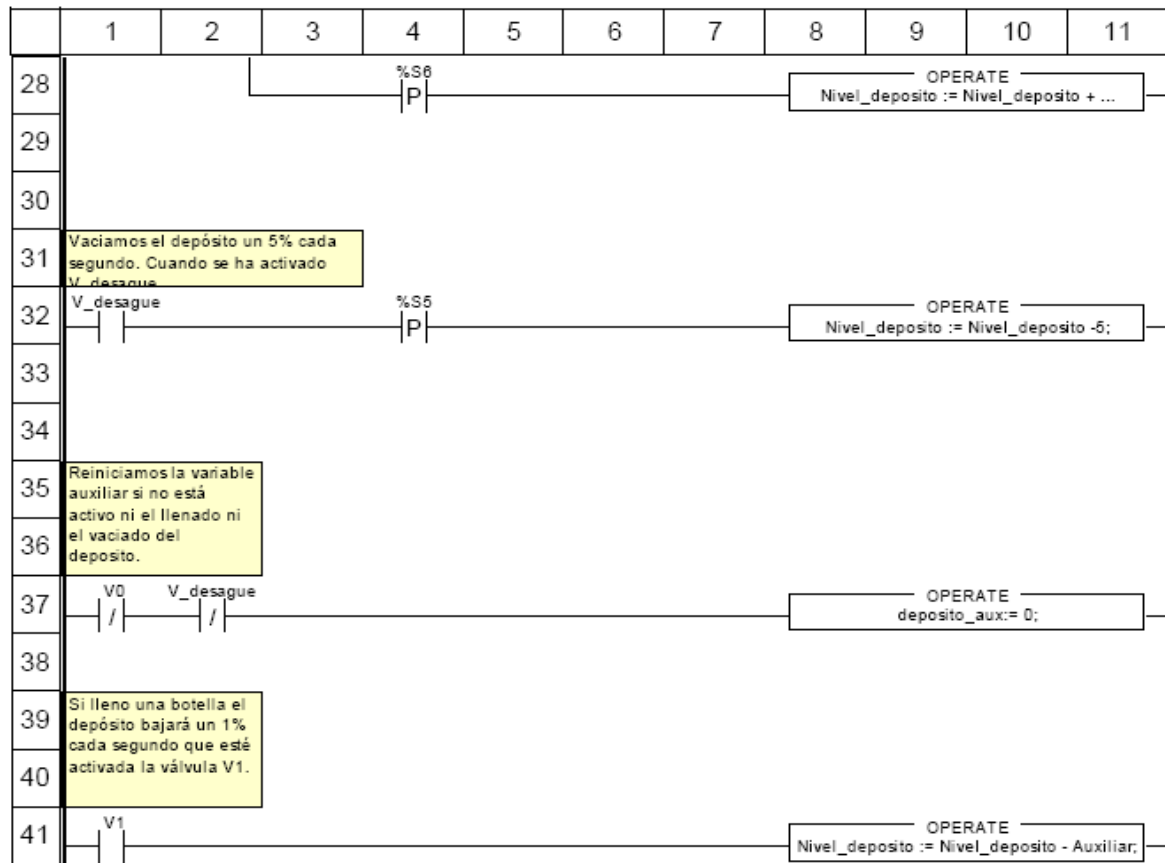
- Schneider Electric: el especialista global en control de la energía. [En línea]*
[1] <<http://www.schneiderelectric.es/sites/spain/es/empresa/perfil/historia/historia-schneider-electric.page>>
[Consulta 1 de Junio de 2013].
- [2] *Schneider Electric: una realidad con 170 años de historia.*
- Rafael L. Ferrer Cutillas. Sistema de automatización de deslastre de cargas eléctricas. Ingeniería Técnica Industrial. Departamento de Ingeniería Electrónica, Eléctrica y Automática. Universitat Rovira i Virgili. Enero 2008.*
- [3]
- Concept User Manual. Volume 1. 840 USE 493 00 eng Version 2.5 - SR2.*
- [4]
- ITpedia: la enciclopedia libre de informática técnica. [En línea]*
[5] <<http://itpedia.wikispaces.com/Lenguaje+de+Programaci%C3%B3n+de+Aut%C3%B3matas>>
[Consulta 2 de Junio de 2013].
- BLANCO, Dolores; MALFAZ, María. Práctica 1 de autómatas programables entorno de programación Unity Pro. Departamento de Ingeniería de Sistemas y Automática Universidad Carlos III. Año 2012.*
- [6]
- Wikipedia, La enciclopedia libre [En línea]*
[7] <<http://es.wikipedia.org/wiki/SCADA>>
[Consulta: 5 de Junio de 2013].
- Crespo Ródenas, Javier. Automatización y diseño Scada con TP177B PN/DP de aplicaciones docentes con Software Scada y autómatas: S7-200 y S7-300. I.T.I Electrónica Industrial. Universidad politécnica de Cartagena. Noviembre de 2008.*
- [8]
- [9], *PL7 PRO, Manual de referencia.*
- [10]
- MUÑOZ HORCAJUELO, María. Simulación de la automatización de procesos con Unity Pro. Departamento de Ingeniería de Sistemas y Automática. Universidad Carlos III de Madrid. Año 2012.*
- [11]

ANEXO I.

Programación Modo Docente.

Tiempos_y_deposito : [MAST]





Etiquetas truncadas:

Etiqueta	Posición(es)
Auxiliar >= 4 AND Auxiliar <= 14	(5, 22)
Nivel deposito = Nivel deposito + (deposito_aux/1000);	(8, 28)

Condiciones : [MAST]

```

1| 10| 20| 30| 40| 50| 60| 70| 80| 90| 100| 110|
2 (*Comienza el proceso*)
3
4
5 IF (Boton_Inicio = TRUE) AND (Nivel_deposito>20) AND Vaciar_deposito = FALSE THEN
6
7
8 (* Tratamos el llenado de la botella *)
9 IF (Posicion1 = TRUE)AND(Posicion2 = FALSE)
10 AND(Posicion3 = FALSE)AND(Posicion4 = FALSE) THEN
11
12 Pinta:=1;
13
14 if (Auxiliar<3) THEN
15 Posicion := 1;
16
17 elsif (Auxiliar>=3 AND Auxiliar<5 AND(V1 = TRUE) ) THEN
18 Posicion := 2;
19
20 elsif (Auxiliar>=5 AND(V1 = TRUE) ) THEN
21 Posicion := 3;
22
23 END_IF;
24 END_IF;
25
26 (* Tratamos el taponado de la botella *)
27 IF (Posicion1 = FALSE)AND(Posicion2 = TRUE)
28 AND(Posicion3 = FALSE)AND(Posicion4 = FALSE) THEN
29
30 Pinta:=2;
31
32 if (Auxiliar<8) THEN
33 Posicion := 5;
34
35 elsif (Taponadora = TRUE)THEN
36 Posicion:= 6;
37 END_IF;
38 END_IF;
39
40
41 (* Tratamos el etiquetado de la botella *)
42 IF (Posicion1 = FALSE)AND(Posicion2 = FALSE)
43 AND(Posicion3 = TRUE)AND(Posicion4 = FALSE) THEN
44
45 Pinta:=3;
46 Posicion:= 8;
47
48 IF (Auxiliar>=14 AND(Posicion_etiquetadora = TRUE)) THEN
49
50 Posicion := 9;
51
52 END_IF;
53 END_IF;
54
55 (* Tratamos el control de calidad de la botella *)
56 IF (Posicion1 = FALSE)AND(Posicion2 = FALSE)
57 AND(Posicion3 = FALSE)AND(Posicion4 = TRUE) THEN
58
59
60 Posicion:= 11;
61
62 IF (Auxiliar>=8) THEN
63
64 IF (Sensor_nivel = TRUE) THEN
65 Posicion := 12;
66 Pinta:=4;
67
68 ELSE
69 Posicion := 14;
70 Pinta:=5;
71

```

```

72 END_IF;
73 END_IF;
74 END_IF;
75
76
77 (* Desplazamiento de la botella a lo largo de la cinta. *)
78 IF (Posicion1 = FALSE)AND(Posicion2 = FALSE)
79 AND(Posicion3 = FALSE)AND(Posicion4 = FALSE)
80 AND(M_Cinta = TRUE OR M_Cinta2 =TRUE) THEN
81
82 Case Pcinta OF
83
84 1: Posicion := 4;
85 2: Posicion := 7;
86 3: Posicion := 10;
87 4: Posicion := 13;84 1: Posicion := 4;
88 5: Posicion := 15;
89
90 else
91 if (Boton_Inicio =TRUE) THEN
92 Posicion:=-1;
93 else
94 Posicion:=0;
95 END_IF;
96 END_CASE;
97 END_IF;
98
99 (* Devuelvo la botella al principio *)
100 IF (Posicion = 13 OR Posicion = 15 AND
101 (Auxiliar>=5 AND Empaquetadora = FALSE AND Alarma = False AND Nivel_deposito>=20 )) THEN
102
103 Posicion :=-1;
104 Pcinta := -1;
105
106 END_IF;
107
108 ELSE (* Sino no se dan las condiciones para iniciar el proceso*)
109 Posicion:=-1;
110
111 END_IF;
112
113

```

ANEXO II.

Documentación técnica del proyecto.

DOCUMENTACIÓN TÉCNICA

Proyecto

Proyecto	Sistema de Embotellado
Diseñador	Álvaro Martínez Galindo
Aplicación	PFG.ALVAROMTZG.EMBOTELLADORA.stu
Versión del software	Unity Pro XL V6.0
Fecha de creación	13/06/2013 05:58:55
Fecha de la última modificación	13/06/2013 05:58:55
PLC de destino	TSX P57 104M 02.6057-1, Programa de 224Kb, Unitelway

Autor:	1 Portada	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 1 - 1/1

Contenido

1 Portada	1 página
2 Contenido	3 páginas
3 Configuración	4 páginas
3.1 0 : Bus X	4 páginas
3.1.1 0 : TSX RKY 6EX	4 páginas
3.1.1.1 0 : TSX P57 104M	1 página
3.1.1.2 1 : TSX DEY 16FK	1 página
3.1.1.3 2 : TSX DSY 16R5	1 página
4 Variables e instancias FB	5 páginas
5 Estructura de la aplicación	4 páginas
6 Programa	65 páginas
6.1 Tareas	65 páginas
6.1.1 MAST	65 páginas
6.1.1.1 Secciones	64 páginas
6.1.1.1.1 Embotelladora	50 páginas
6.1.1.1.1.1 Chart	5 páginas
6.1.1.1.1.2 Acciones	28 páginas
6.1.1.1.1.2.1 Botella0	1 página
6.1.1.1.1.2.2 Botella1	1 página
6.1.1.1.1.2.3 Botella2	1 página
6.1.1.1.1.2.4 Botella3	1 página
6.1.1.1.1.2.5 Botella4	1 página
6.1.1.1.1.2.6 Botella5	1 página
6.1.1.1.1.2.7 Botella6	1 página
6.1.1.1.1.2.8 Decrementar_Deposito	1 página
6.1.1.1.1.2.9 Botella8	1 página
6.1.1.1.1.2.10 Proceso_etiquetado	1 página
6.1.1.1.1.2.11 Maq_etiquetadora	1 página
6.1.1.1.1.2.12 Etiquetado_Time	1 página
6.1.1.1.1.2.13 Botella10	1 página
6.1.1.1.1.2.14 Botella7	1 página
6.1.1.1.1.2.15 Botella9	1 página
6.1.1.1.1.2.16 Botella11	1 página
6.1.1.1.1.2.17 Botella12	1 página
6.1.1.1.1.2.18 Botella13	1 página
6.1.1.1.1.2.19 Sin_botella	1 página
6.1.1.1.1.2.20 Botella14	1 página
6.1.1.1.1.2.21 Botella15	1 página

Contenido

6.1.1.1.1.2.22 Botella16	1 página
6.1.1.1.1.2.23 Reiniciar_contador	1 página
6.1.1.1.1.2.24 Aumeta_cont_des	1 página
6.1.1.1.1.2.25 Aumenta_cont_op	1 página
6.1.1.1.1.2.26 Aumenta_paquetes	1 página
6.1.1.1.1.2.27 Vaciado_BotellasDesechadas	1 página
6.1.1.1.1.2.28 Reiniciar_T_aux	1 página
6.1.1.1.1.3 Transiciones	16 páginas
6.1.1.1.1.3.1 B_inicio	1 página
6.1.1.1.1.3.2 P1	1 página
6.1.1.1.1.3.3 Tiempo_Llenado	1 página
6.1.1.1.1.3.4 P2	1 página
6.1.1.1.1.3.5 Tiempo_tapon	1 página
6.1.1.1.1.3.6 P3	1 página
6.1.1.1.1.3.7 Control_Defectuoso	1 página
6.1.1.1.1.3.8 Tiempo_Etiquetado	1 página
6.1.1.1.1.3.9 P4	1 página
6.1.1.1.1.3.10 Optima	1 página
6.1.1.1.1.3.11 Empaquetado_SI	1 página
6.1.1.1.1.3.12 Empaquetado_NO	1 página
6.1.1.1.1.3.13 Tiempo_Empaquetado	1 página
6.1.1.1.1.3.14 Tiempo_desecho	1 página
6.1.1.1.1.3.15 Dep_des_lleno	1 página
6.1.1.1.1.3.16 Problema_solucionado	1 página
6.1.1.1.2 Control_deposito	11 páginas
6.1.1.1.2.1 Chart	2 páginas
6.1.1.1.2.2 Acciones	2 páginas
6.1.1.1.2.2.1 Llenar_tanque	1 página
6.1.1.1.2.2.2 vaciar_tanque	1 página
6.1.1.1.2.3 Transiciones	6 páginas
6.1.1.1.2.3.1 Lleno	1 página
6.1.1.1.2.3.2 Tanque_vacio	1 página
6.1.1.1.2.3.3 Desague	1 página
6.1.1.1.2.3.4 No_vacio	1 página
6.1.1.1.2.3.5 vacio	1 página
6.1.1.1.2.3.6 No_Lleno	1 página
6.1.1.1.3 Parada_de_Emergencia	2 páginas
6.1.1.1.4 Reset_contadores	1 página
7 Tablas de animación	2 páginas
7.1 Entradas	1 página
7.2 Salidas	1 página

Contenido

8 Pantallas de operador	1 página
8.1 Instalacion	1 página
9 Movimiento	1 página
10 Referencias cruzadas	7 páginas
11 Objetos de usuario	1 página

Total: 94 páginas

Autor:	2 Contenido	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 2 - 3/3

0 : TSX RKY 6EX

Slot	Familia	Referencia
(P)	Alimentación	TSX PSY 2600M
0	Premium	TSX P57 104M
1	Binario	TSX DEY 16FK
2	Binario	TSX DSY 16R5

Autor:	3.1 0 : Bus X	Impreso el 24/06/2013
Dept.:	3.1.1 0 : TSX RKY 6EX	
Proyecto: Sistema de Embotellado		Página: 3.1.1 - 1/4

Este documento es propiedad de XXX y no se puede reproducir ni comercializar sin autorización previa.

0.0 : TSX P57 104M

Identificación del módulo:

Referencia comercial : TSX P57 104M Designación : 57-1, Programa de 224Kb, Unitelway
Dirección : 0.0 Símbolo :

Característica de memoria

Nombre de la tarjeta de memoria : Ninguna

Modalidad de servicio

Entrada de ejecución/detención : No
Protección de memoria : No
Iniciar ejecución automática : No
Resetear MWi : Sí
Sólo arranque en frío : No

Datos

Cantidad de bits : 256
Cantidad de palabras : 512
Cantidad de constantes : 128
Cantidad de bits de sistema : 128
Cantidad de palabras de sistema : 168

Canal 0 :

Función específica de la aplicación : Conexión Uni-Telway
Tipo de canales de E/S : Canal integrado
Tarea : MAST
Tipo : Maestro
Velocidad de transmisión : 19.200 bits/s Datos : a 8 bits
Detener : a 1 bit Paridad : Impar
Cantidad de slaves : 8
Valor de timeout en MS : 30

Canal 100 :

Función específica de la aplicación : Ninguno
Tipo de canales de E/S : Canal integrado

0.1 : TSX DEY 16FK

Identificación del módulo:

Referencia comercial : TSX DEY 16FK
Dirección : 0.1

Designación : 16E RAPID 24 VCC SINK CON.
Símbolo :

Parámetros comunes [0-7]

Monitorización de alimentación : Activo
Tarea : MAST

Parámetros del canal de entrada [0-7]

Canal	Dirección	Símbolo	Filtro	Función
0	%I0.1.0.0	RESET	4 ms	
1	%I0.1.1.0	Boton_Inicio	4 ms	
2	%I0.1.2.0	Continuar_proceso	4 ms	
3	%I0.1.3.0	Parada_emergencia	4 ms	
4	%I0.1.4.0	Vaciar_deposito	4 ms	
5	%I0.1.5.0	Posicion1	4 ms	
6	%I0.1.6.0	Posicion2	4 ms	
7	%I0.1.7.0	Posicion3	4 ms	

Parámetros comunes [8-15]

Monitorización de alimentación : Activo
Tarea : MAST

Parámetros del canal de entrada [8-15]

Canal	Dirección	Símbolo	Filtro	Función
8			4 ms	
9			4 ms	
10			4 ms	
11			4 ms	
12			4 ms	
13			4 ms	
14			4 ms	
15			4 ms	

Autor:	3.1.1 0 : TSX RKY 6EX	Impreso el 24/06/2013
Dept.:	3.1.1.2 1 : TSX DEY 16FK	
Proyecto: Sistema de Embotellado		Página: 3.1.1.2 - 1/1

Este documento es propiedad de XXX y no se puede reproducir ni comercializar sin autorización previa.

0.2 : TSX DSY 16R5

Identificación del módulo:

Referencia comercial : TSX DSY 16R5
Dirección : 0.2

Designación : 16S RELÉ 50 VA, BL.TER.
Símbolo :

Parámetros comunes [0-7]

Tarea : MAST
Modalidad de retorno : Retorno

Parámetros del canal de salida [0-7]

Canal	Dirección	Símbolo	Valor de ret.
0	%Q0.2.0.0	V0	0
1	%Q0.2.1.0	V1	0
2	%Q0.2.2.0	V_desague	0
3	%Q0.2.3.0	M_Cinta	0
4	%Q0.2.4.0	M_Cinta2	0
5	%Q0.2.5.0	Taponadora	0
6	%Q0.2.6.0	Maquina_etiquetado	0
7	%Q0.2.7.0	Posicion_etiquetadora	0

Parámetros comunes [8-15]

Tarea : MAST
Modalidad de retorno : Retorno

Parámetros del canal de salida [8-15]

Canal	Dirección	Símbolo	Valor de ret.
8	%Q0.2.8.0	Empaquetadora	0
9	%Q0.2.9.0	Alarma	0
10	%Q0.2.10.0		0
11	%Q0.2.11.0		0
12	%Q0.2.12.0		0
13	%Q0.2.13.0		0
14	%Q0.2.14.0		0
15	%Q0.2.15.0		0

Variables e instancias FB

BOOL

Nombre	Const	Dirección	Comentario	Valor	Utilizado	DG
B_inicio	NO				2	NO
Control Defectuoso	NO				2	NO
Dep_des lleno	NO				2	NO
Desague	NO				2	NO
Empaquetado_NO	NO				2	NO
Empaquetado_SI	NO				2	NO
Lleno	NO				2	NO
No Lleno	NO				2	NO
No vacio	NO				2	NO
NOT_Parada_emergencia	NO		NOT(Parada_Emergencia). Condición para que se produzcan las transiciones en los SFC.		5	NO
Optima	NO				2	NO
P1	NO				2	NO
P2	NO				2	NO
P3	NO				2	NO
P4	NO				2	NO
Problema_solucionado	NO				2	NO
Tanque vacio	NO				2	NO
Tiempo desecho	NO				2	NO
Tiempo_Empaquetado	NO				2	NO
Tiempo Etiquetado	NO				2	NO
Tiempo Llenado	NO				2	NO
Tiempo tapon	NO				2	NO
Vacio	NO				2	NO

EBOOL

Nombre	Const	Dirección	Comentario	Valor	Utilizado	DG
Alarma	NO		Señal acústica que se activa en caso de que se detecte algún problema.		5	NO
Boton Inicio	NO		Boton para iniciar el proceso.		7	NO
Continuar_proceso	NO		Botón que se pulsará para indicar que se ha subsanado cualquier error.		3	NO
Empaquetadora	NO		Máquina encargada del empaquetado de botellas.		7	NO
M Cinta	NO		Motor que mueve la cinta transportadora principal.		17	NO
M Cinta2	NO		Cinta transportadora que deshecha las botellas malas,		10	NO
Maquina_etiquetado	NO		Máquina para el etiquetado de las botellas.		12	NO
N20	NO		Sensor de nivel que indica que el tanque está por encima del 20%		3	NO
N100	NO		Sensor de nivel que indica que el tanque esta lleno.		3	NO
Parada emergencia	NO		Pulsador de Parada de emergencia.		6	NO
Posicion1	NO		Sensor que indica que la botella ha llegado a la zona de llenado.		9	NO
Posicion2	NO		Sensor que indica que la botella esta en la zona de taponado.		9	NO
Posicion3	NO		Sensor que indica que la botella esta en la posición de etiquetado.		9	NO
Posicion4	NO		Sensor que indica que la botella ha llegado a la zona de control de calidad.		10	NO
Posicion_etiquetadora	NO		Motor para mover la posición de la máquina de etiquetado. 1 Abajo, 0 Arriba.		5	NO
RESET	NO		Boton Reset.		3	NO
Sensor_nivel	NO		Sensor que indica si la botella contiene el liquido mínimo para pasar el control de calidad.		7	NO
Taponadora	NO		Proceso de colocar el tapón en la botella.		5	NO

Autor:	4 Variables e instancias FB	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 4 - 1/5

Variables e instancias FB

Nombre	Const	Dirección	Comentario	Valor	Utilizado	DG
V0	NO		Válvula que llena el tanque que contiene el fluido.		8	NO
V1	NO		Válvula que abre el tanque que contiene el fluido.		5	NO
V_desague	NO		Válvula que destina el contenido del tanque hacia el desagüe.		6	NO
Vaciar_deposito	NO		Botón utilizado para vaciar el contenido del tanque.		5	NO

INT

Nombre	Const	Dirección	Comentario	Valor	Utilizado	DG
Contador_botellas	NO		Cuenta las botellas llenadas correctamente.		7	NO
Contador_botellas2	NO		Cuenta el número de botellas desechadas.		10	NO
Contador_Paquetes	NO		Cuenta el número de paquetes que se han realizado hasta el momento.		4	NO
Nivel_deposito	NO		Indicará el nivel de llenado del depósito.		19	NO
Posicion	NO		Indicará el estado de la botella y el gráfico que debe aparecer.		33	NO
T_aux2	NO		Almacena como entero el tiempo de T aux.		5	NO
Tiempo_etiquetador_a	NO		Muestra el tiempo que resta para que acabe el proceso de etiquetado.		3	NO
Tiempo_etiquetador_a2	NO		Almacena como entero el tiempo del estado etiquetado.		4	NO

MC_STOP

Nombre	Comentario	Valor	Utilizado	DG
MC_STOP_1			0	
<entradas>				
AXIS	Axis designed to work with.			
AxisReady	Communication with the Axis is OK			
AxisType	L17 : 1, L15: 2, IFX : 3, ATV31 : 304, ATV71 : 305, L05: 306, L32: 407			
AxisReference	Reference of the drive			
PLCTask	Identification of the PLC task (1:MAST, 2:FAST)			
NetworkType	Defines the network type (1:CANOpen, 2:Ethernet)			
AxisMajorVersion	Firmware minor version of the drive, filled by AMT			
AxisMinorVersion	Firmware minor version of the drive, filled by AMT			
AxisMajorVersionRead	Firmware major version, read from drive			
AxisMinorVersionRead	Firmware minor version, read from drive			
EXECUTE	Execution starts on front edge			
<salidas>				
ERROR	Block execution has failed			
DONE	Execution done wiBlock has finished its job successfullythout errors			
BUSY	Block is busy working			
ERRORID	Error code to explain why block execution has failed			
COMMANDABORTED	Block execution has been aborted due to another block taking control			
<público>				
LOG_FILTER	Apply a filter before logging	255		

SFCCHART STATE

Nombre	Const	Dirección	Comentario	Utilizado
Control_deposito	NO			0
Embotelladora	NO			0

SFCCNTRL

Nombre	Comentario	Valor	Utilizado	DG
SFCCNTRL_1			0	
<entradas>				
CHARTREF	Link to SFC-chart			
INIT	reset and start a SFC Chart			
CLEAR	reset a SFC Chart			
DISTIME	disable time supervision			

Autor:	4 Variables e instancias FB	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 4 - 2/5

Variables e instancias FB

Nombre	Comentario	Valor	Utilizado	DG
DISTRANS	disable transition processing			
DISACT	disable action processing			
STEPUN	Activate next step independent of transition			
STEPDEP	Dependent on transition activate next step			
RESETERR	Clear supervision time violation indication			
DISRMOTE	Disallow remote operation			
ALLTRANS	Solve all transition sections			
RESSTEPT	Reset elapsed time of all steps			
<salidas>				
INITST	Show value of INIT in effect			
CLEARST	Show value of CLEAR in effect			
TIMEDIS	show value of DISTIME in effect			
TRANSDIS	Show value of DISTRANS in effect			
ACTDIS	Show value of DISACT in effect			
MODECHG	Report operation mode changes			
STATECHG	Report state changes			
TIMEERR	Report supervision time violation			
TERRACT	Error in supervision time violation			

SFCSTEP STATE

Nombre	Const	Dirección	Comentario	Utilizado
Cinta_1	NO			1
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Cinta_2	NO			2
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Cinta_3	NO			2
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Cinta_4	NO			2
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Cinta_5	NO			5
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Control_calidad	NO			4
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Deposito_des_Lleno	NO			2
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Desecho	NO			5
t	NO			
x	NO			

Variables e instancias FB

Nombre	Const	Dirección	Comentario	Utilizado
tminErr	NO			
tmaxErr	NO			
Empaquetado	NO			2
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Etiquetado	NO			9
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Inicio	NO			2
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Llenado	NO			3
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Rellenar	NO			3
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Reposo Tanque	NO			1
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Tapa	NO			2
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			
Vaciado	NO			3
t	NO			
x	NO			
tminErr	NO			
tmaxErr	NO			

TIME

Nombre	Const	Dirección	Comentario	Valor	Utilizado	DG
T_aux	NO		Guarda el tiempo que está activa la parada de emergencia.		31	NO

TOF

Nombre	Comentario	Valor	Utilizado	DG
TOF 0			0	
<entradas>				
IN	Start delay			
PT	Preset delay time			
<salidas>				
Q	Delayed output			
ET	Internal time			
TOF 1			0	
<entradas>				

Autor:	4 Variables e instancias FB	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 4 - 4/5

Variables e instancias FB

Nombre	Comentario	Valor	Utilizado	DG
IN	Start delay			
PT	Preset delay time			
<salidas>				
Q	Delayed output			
ET	Internal time			

Estructura de la aplicación

VISTA ESTRUCTURAL

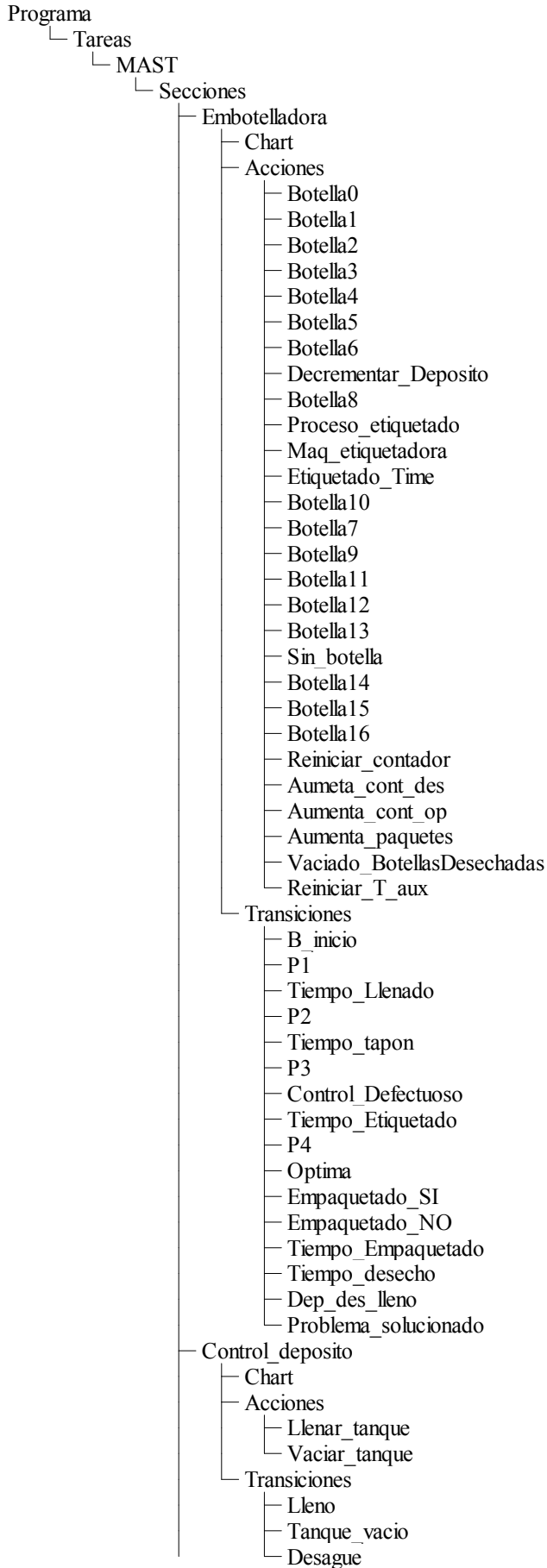
SECCIÓN	CONDICIÓN DE VALIDACIÓN	COMENTARIO DE SECCIÓN	MÓDULO	LENGUAJE
Embotelladora	NOT_Parada_emergencia			SFC
Chart				SFC
Botella0				LD
Botella1				LD
Botella2				LD
Botella3				LD
Botella4				LD
Botella5				LD
Botella6				LD
Decrementar_Deposito				LD
Botella8				LD
Proceso_etiquetado				LD
Maq_etiquetadora				LD
Etiquetado_Time				LD
Botella10				LD
Botella7				LD
Botella9				LD
Botella11				LD
Botella12				LD
Botella13				LD
Sin_botella				LD
Botella14				LD
Botella15				LD
Botella16				LD
Reiniciar_contador				LD
Aumeta_cont_des				LD
Aumenta_cont_op				LD
Aumenta_paquetes				LD
Vaciado_BotellasDesechadas				LD
Reiniciar_T_aux				LD
B_inicio				LD
P1				LD
Tiempo_Llenado				LD
P2				LD
Tiempo_tapon				LD
P3				LD
Control_Defectuoso				LD
Tiempo_Etiquetado				LD
P4				LD
Optima				LD
Empaquetado_SI				LD
Empaquetado_NO				LD
Tiempo_Empaquetado				LD
Tiempo_desecho				LD
Dep_des_lleno				LD
Problema_solucionado				LD
Control_deposito	NOT_Parada_emergencia			SFC
Chart				SFC
Llenar_tanque				LD
Vaciar_tanque				LD
Lleno				LD
Tanque_vacio				LD
Desague				LD
No_vacio				LD
Vacio				LD

Estructura de la aplicación

SECCIÓN	CONDICIÓN DE VALIDACIÓN	COMENTARIO DE SECCIÓN	MÓDULO	LENGUAJE
No Lleno				LD
Parada de Emergencia				LD
Reset_contadores				LD

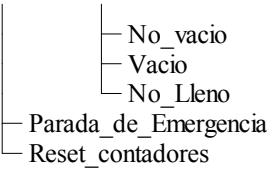
Estructura de la aplicación

CALL TREE



Autor:	5 Estructura de la aplicación	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 5 - 3/4

Estructura de la aplicación



MAST

Propiedades específicas

Configuración	Cíclica
Configuración del periodo de tareas	0
Configuración del tiempo de watchdog	250

Autor:	6.1 Tareas 6.1.1 MAST	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 6.1.1 - 1/65

Embotelladora : [MAST]

Comentario

Propiedades comunes

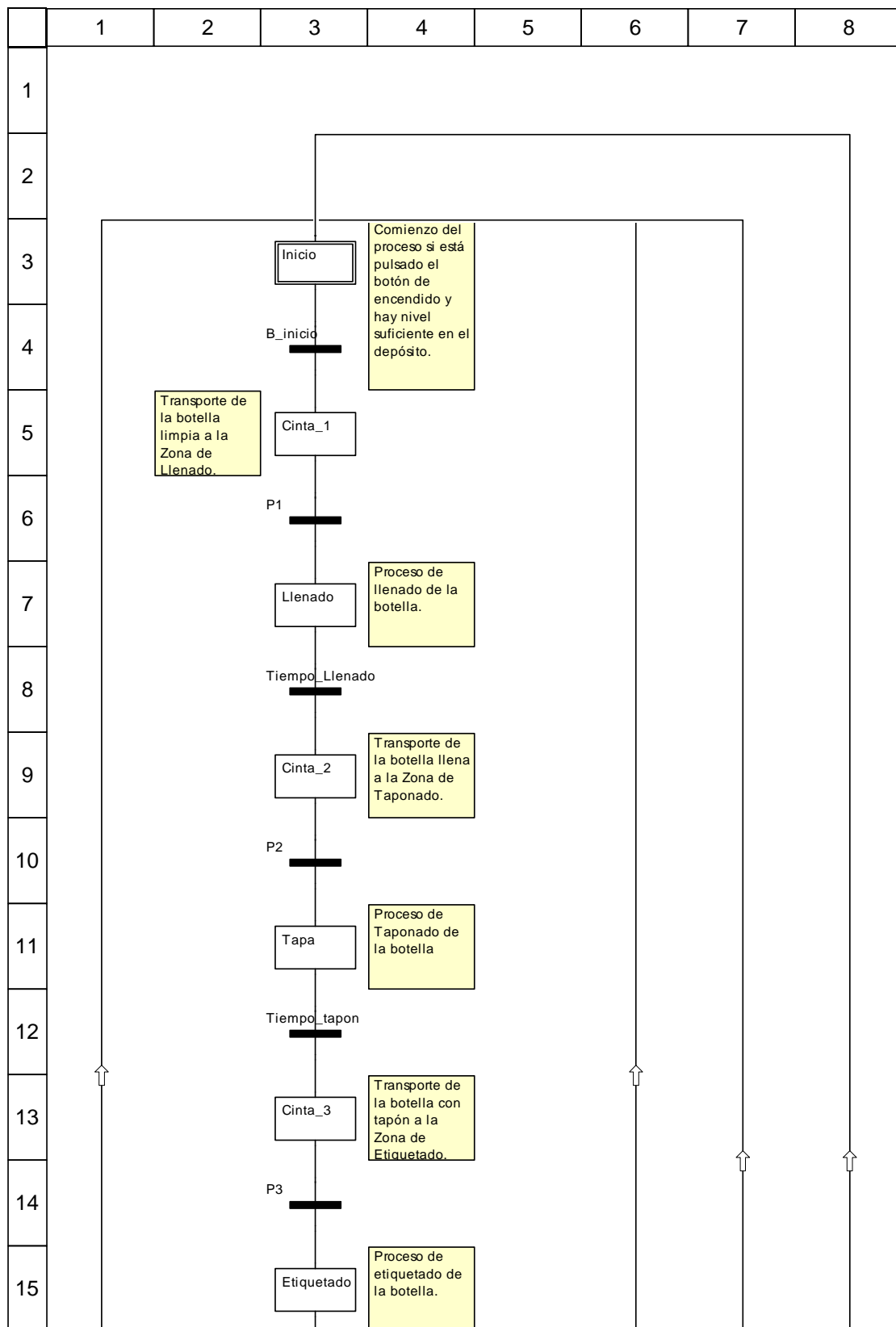
Módulo funcional	
Nombre de la condición	NOT Parada emergencia

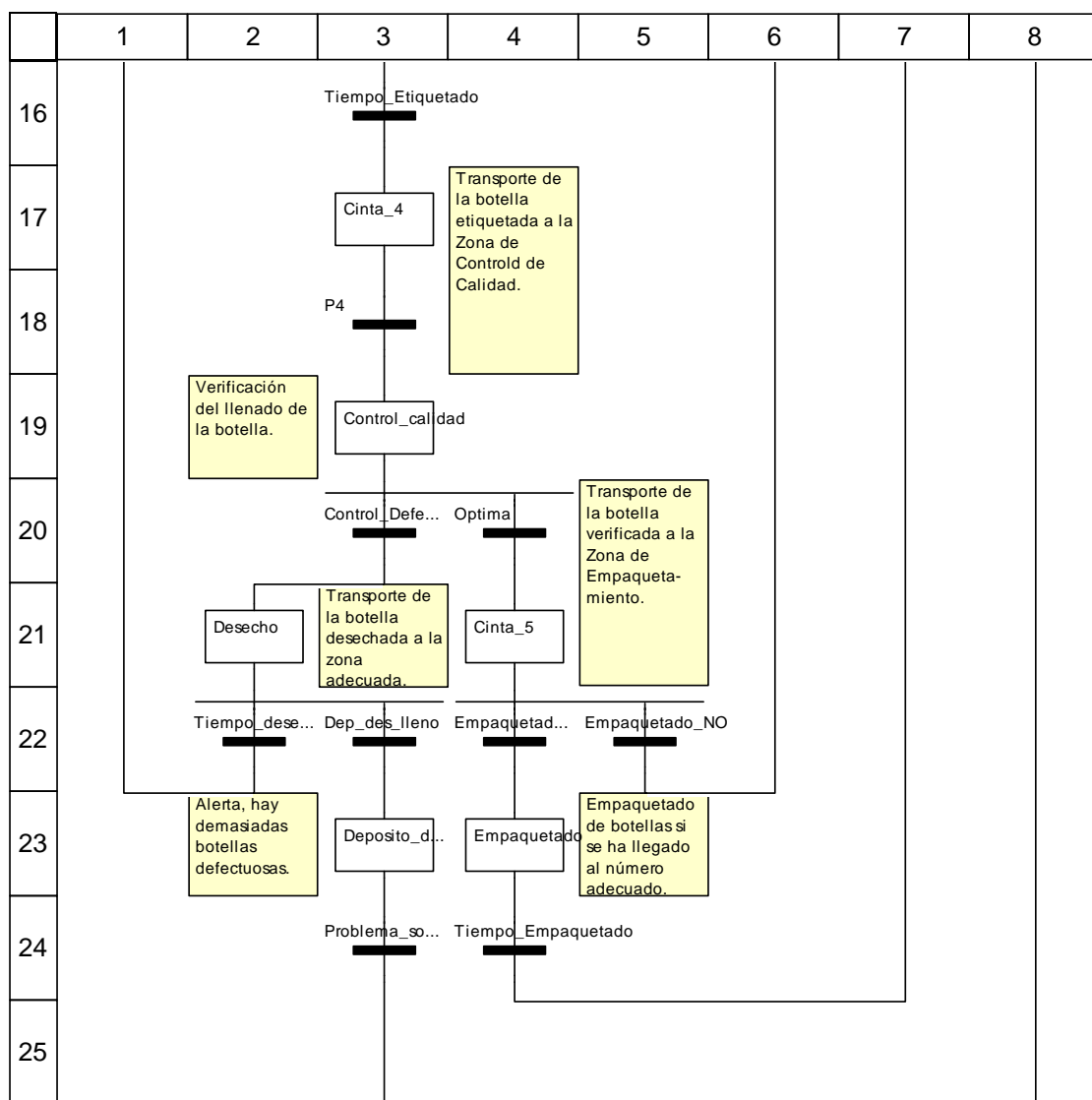
Propiedades específicas

Control de operador	No
Número de área	0

Autor:	6.1.1.1 Secciones 6.1.1.1.1 Embotelladora	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 6.1.1.1.1 - 1/50

Chart : [MAST - Embotelladora]





Descripción de objeto

Pasos:

Cinta 1			(3, 5)
Tiempo de supervisión mín./máx.:			Tiempo de retardo:
Comentario:			
Acciones:			
Descriptor: N	Tiempo:	Variable: M Cinta	
Descriptor: P1	Tiempo:	Sección: LD :: Botella0	
Descriptor: P0	Tiempo:	Sección: LD :: Botella1	
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux	

Cinta 2			(3, 9)
Tiempo de supervisión mín./máx.:			Tiempo de retardo:
Comentario:			
Acciones:			
Descriptor: N	Tiempo:	Variable: M Cinta	
Descriptor: N	Tiempo:	Sección: LD :: Botella4	

Autor:	6.1.1.1.1 Embotelladora 6.1.1.1.1 Chart	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 6.1.1.1.1 - 2/5

Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux
-------------------	---------	--------------------------------

Cinta 3		(3, 13)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Variable: M Cinta
Descriptor: N	Tiempo:	Sección: LD :: Botella7
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux

Cinta_4		(3, 17)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Variable: M Cinta
Descriptor: N	Tiempo:	Sección: LD :: Botella10
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux

Cinta 5		(4, 21)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Variable: M Cinta
Descriptor: N	Tiempo:	Sección: LD :: Botella12
Descriptor: N	Tiempo:	Sección: LD :: Botella13
Descriptor: P1	Tiempo:	Sección: LD :: Aumenta_cont_op
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux

Control calidad		(3, 19)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Sección: LD :: Botella11
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux

Deposito des Lleno		(3, 23)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Variable: Alarma
Descriptor: P0	Tiempo:	Sección: LD :: Vaciado_BotellasDesechadas
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux

Desecho		(2, 21)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Variable: M Cinta2
Descriptor: N	Tiempo:	Sección: LD :: Botella14
Descriptor: N	Tiempo:	Sección: LD :: Botella15
Descriptor: P1	Tiempo:	Sección: LD :: Aumeta_cont_des

Autor:	6.1.1.1.1 Embotelladora	Impreso el 24/06/2013
Dept.:	6.1.1.1.1.1 Chart	
Proyecto: Sistema de Embotellado		Página: 6.1.1.1.1 - 3/5

Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux
-------------------	---------	--------------------------------

Empaquetado		(4, 23)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Variable: Empaquetadora
Descriptor: P1	Tiempo:	Sección: LD :: Botella16
Descriptor: P0	Tiempo:	Sección: LD :: Reiniciar_contador
Descriptor: P0	Tiempo:	Sección: LD :: Aumenta_paquetes
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux

Etiquetado		(3, 15)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: P1	Tiempo:	Sección: LD :: Botella8
Descriptor: N	Tiempo:	Sección: LD :: Proceso_etiquetado
Descriptor: N	Tiempo:	Sección: LD :: Maq_etiquetadora
Descriptor: N	Tiempo:	Sección: LD :: Etiquetado_Time
Descriptor: N	Tiempo:	Sección: LD :: Botella9
Descriptor: P0	Tiempo:	Variable: Maquina_etiquetado
Descriptor: P0	Tiempo:	Variable: Posicion_etiquetadora
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux

Inicio (paso inicial)		(3, 3)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: P1	Tiempo:	Sección: LD :: Sin_botella
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux

Llenado		(3, 7)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Sección: LD :: Botella2
Descriptor: N	Tiempo:	Variable: V1
Descriptor: P0	Tiempo:	Sección: LD :: Botella3
Descriptor: P0	Tiempo:	Sección: LD :: Decrementar_Deposito
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux

Tapa		(3, 11)
Tiempo de supervisión mín./máx.:		Tiempo de retardo:
Comentario:		
Acciones:		
Descriptor: N	Tiempo:	Variable: Taponadora
Descriptor: P1	Tiempo:	Sección: LD :: Botella5

Autor:	6.1.1.1.1 Embotelladora 6.1.1.1.1.1 Chart	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 6.1.1.1.1 - 4/5

Descriptor: P0	Tiempo:	Sección: LD :: Botella6
Descriptor: P1	Tiempo:	Sección: LD :: Reiniciar_T_aux

Transiciones:

Nombre	Tipo de condición	Posición	Comentario
LD :: B_inicio	Sección	(3, 4)	
LD :: Control_Defectuoso	Sección	(3, 20)	
LD :: Dep_des_lleno	Sección	(3, 22)	
LD :: Empaquetado_NO	Sección	(5, 22)	
LD :: Empaquetado_SI	Sección	(4, 22)	
LD :: Optima	Sección	(4, 20)	
LD :: P1	Sección	(3, 6)	
LD :: P2	Sección	(3, 10)	
LD :: P3	Sección	(3, 14)	
LD :: P4	Sección	(3, 18)	
LD :: Problema_solucionado	Sección	(3, 24)	
LD :: Tiempo_Empaquetado	Sección	(4, 24)	
LD :: Tiempo_Etiquetado	Sección	(3, 16)	
LD :: Tiempo_Llenado	Sección	(3, 8)	
LD :: Tiempo_desecho	Sección	(2, 22)	
LD :: Tiempo_tapon	Sección	(3, 12)	

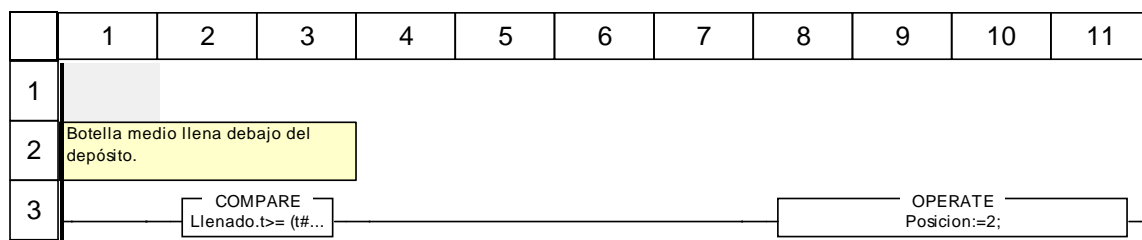
Botella0 <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2	Posición de la botella en el inicio de la cinta transportadora.										
3	<div>OPERATE</div> <div>Posicion:=0;</div>										

Botella1 <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2	Botella vacía, debajo del tanque de depósito.										
3	OPERATE Posicion:=1;										

Botella2 <Acción> : [MAST - Embotelladora]



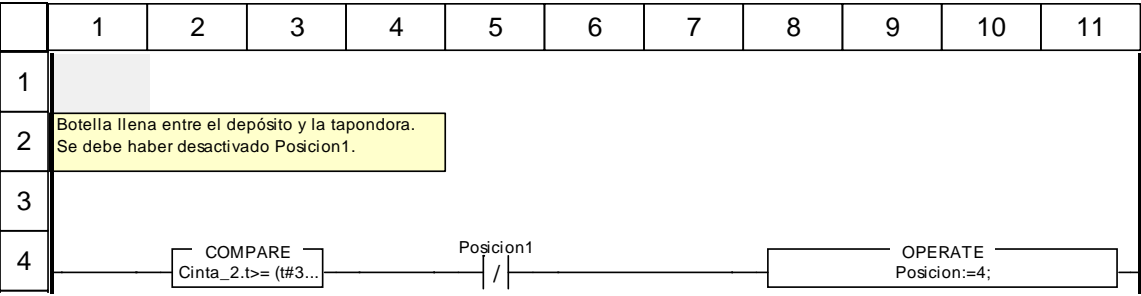
Etiquetas truncadas:

Etiqueta	Posición(es)
Llenado.t>= (t#3s + T_aux)	(2, 3)

Botella3 <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2	Botella llena debajo del depósito.										
3	OPERATE Posicion:=3;										

Botella4 <Acción> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Cinta 2.t >= (t#3s + T_aux)	(2, 4)

Botella5 <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2	Botella llena debajo de la taponadora.										
3								OPERATE Posicion:=5;			

Botella6 <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3	Botella tapondada debajo de la taponadora.										
4								OPERATE Posicion:=6;			

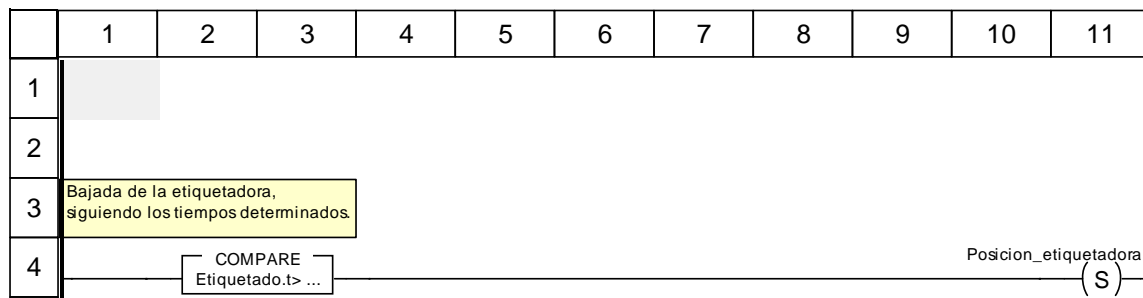
Decrementar_Deposito <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2	Decrementamos el depósito en un 5%.										
3	OPERATE Nivel_deposito:=Nivel_deposito-5;										

Botella8 <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2	Botella taponada sin etiquetar debajo de la etiquetadora.										
3	OPERATE Posicion:=8;										

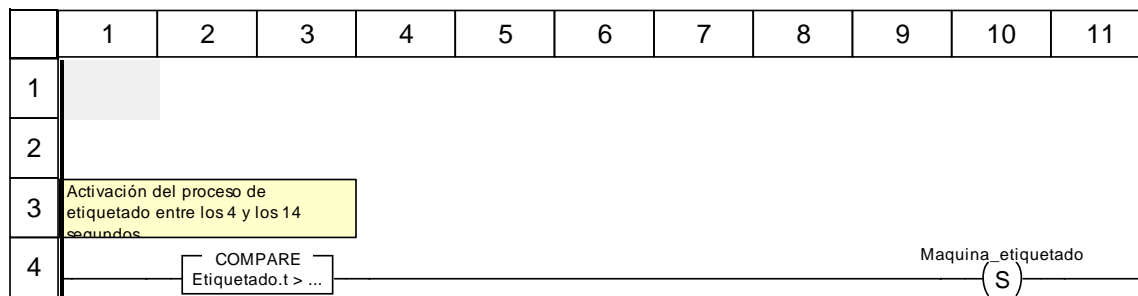
Proceso_etiquetado <Acción> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Etiquetado.t > (t#2s + T aux) AND Etiquetado.t < (t#16s + T aux)	(2, 4)

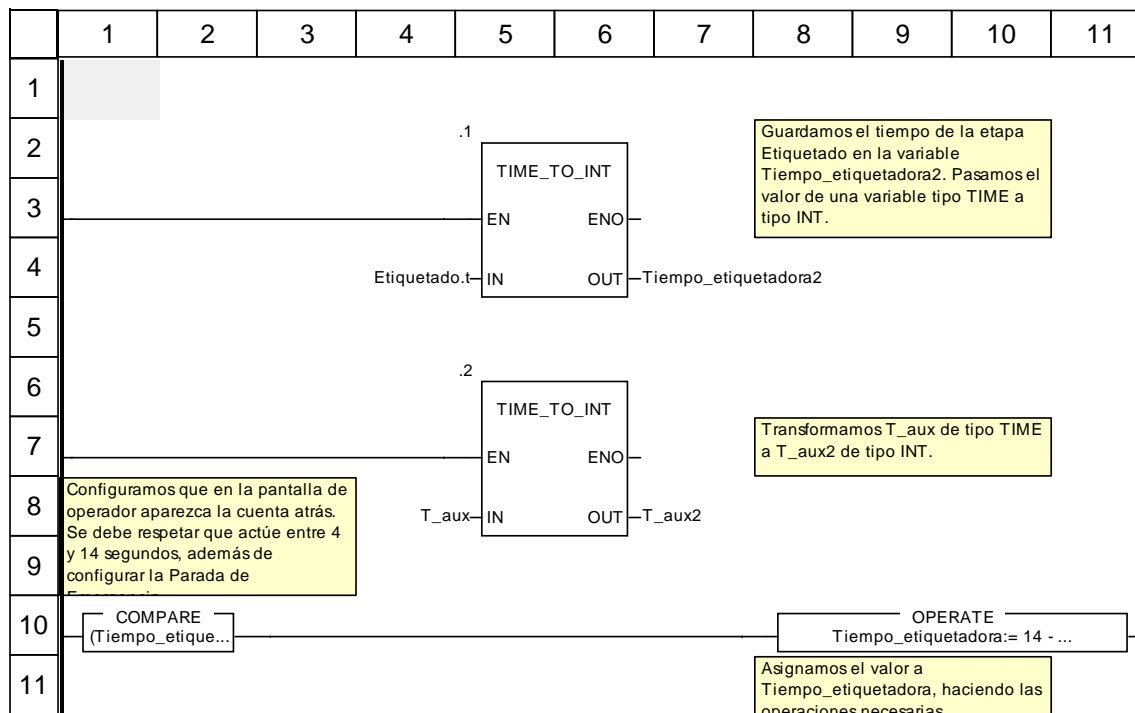
Maq_etiquetadora <Acción> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
$\text{Etiquetado.t} > (t\#4s + T_{\text{aux}}) \text{ AND } \text{Etiquetado.t} < (t\#14s + T_{\text{aux}})$	(2, 4)

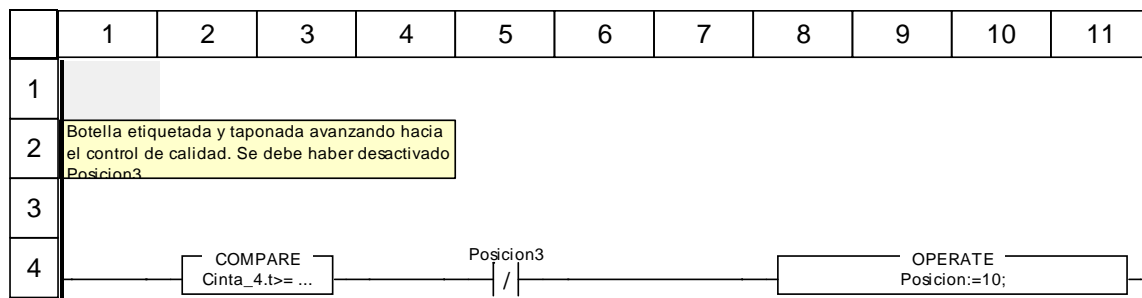
Etiquetado_Time <Acción> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
$(\text{Tiempo_etiquetadora2} - \text{T_aux2}) \geq 4000 \text{ AND } (\text{Tiempo_etiquetadora2} - \text{T_aux2}) \leq 14000$	(1, 10)
$\text{Tiempo_etiquetadora} = 14 - ((\text{Tiempo_etiquetadora2} - \text{T_aux2}) / 1000);$	(8, 10)

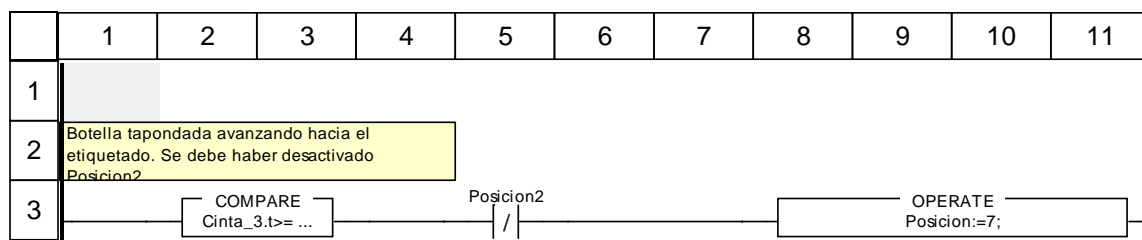
Botella10 <Acción> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Cinta_4.t>= (t#3s+ T_aux)	(2, 4)

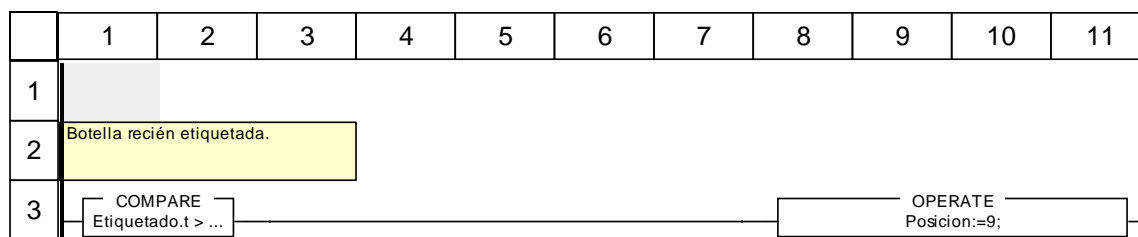
Botella7 <Acción> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Cinta_3.t>= (t#3s+ T_aux)	(2, 3)

Botella9 <Acción> : [MAST - Embotelladora]



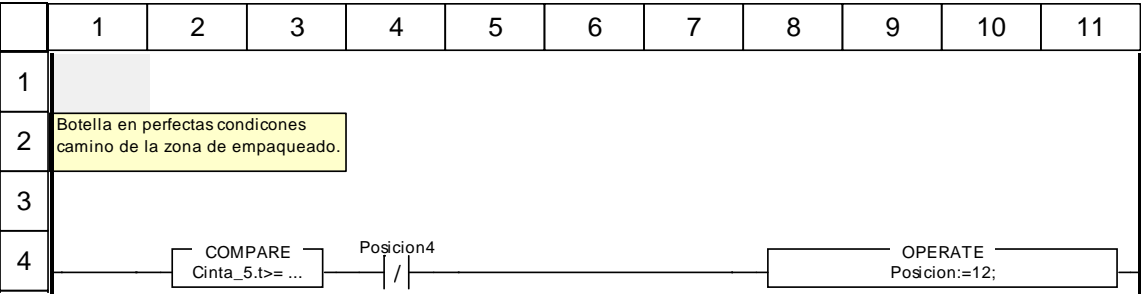
Etiquetas truncadas:

Etiqueta	Posición(es)
Etiquetado.t > (t#2s + T_aux) AND (Etiquetado.t < (t#16s + T_aux))	(1, 3)

Botella11 <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2	Botella en la zona de control de calidad.										
3								OPERATE Posicion:=11;			

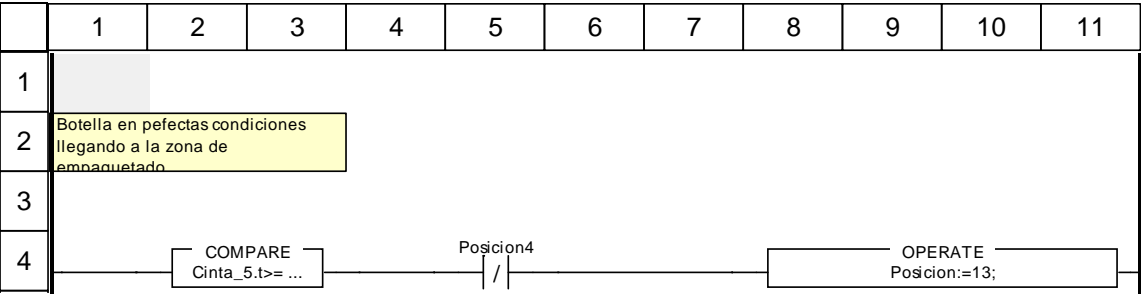
Botella12 <Acción> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Cinta_5.t>= (t#3s+ T_aux)	(2, 4)

Botella13 <Acción> : [MAST - Embotelladora]



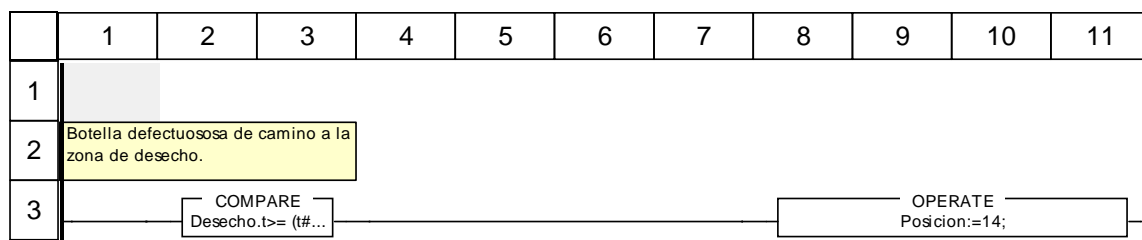
Etiquetas truncadas:

Etiqueta	Posición(es)
Cinta_5.t>= (t#7s+ T_aux)	(2, 4)

Sin_botella <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2	Sin botella en el proceso.										
3								OPERATE Posicion:=-1;			

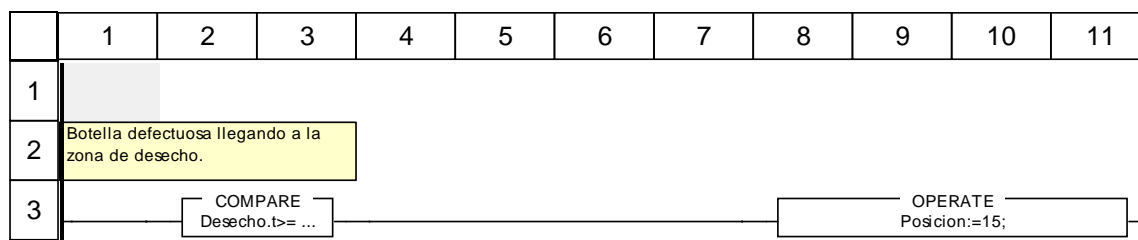
Botella14 <Acción> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Desecho.t >= (t#3s + T_aux)	(2, 3)

Botella15 <Acción> : [MAST - Embotelladora]



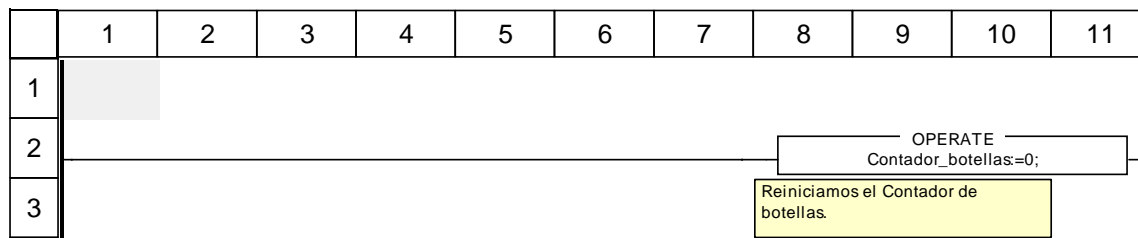
Etiquetas truncadas:

Etiqueta	Posición(es)
Desecho.t>= (t#7s+ T_aux)	(2, 3)

Botella16 <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2	Sin botella en el sistema.										
3	OPERATE Posicion:=16;										

Reiniciar_contador <Acción> : [MAST - Embotelladora]



Aumeta_cont_des <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3											

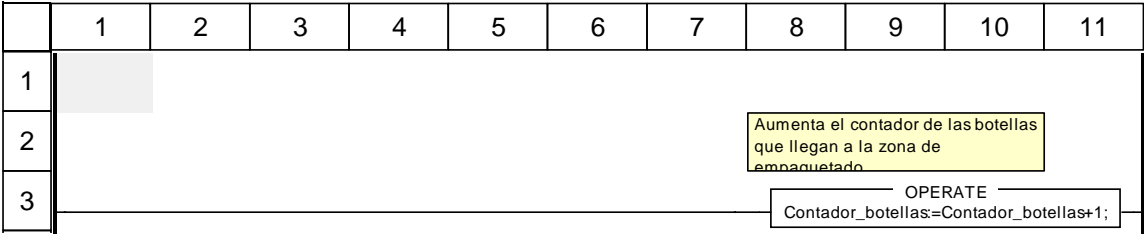
Aumento del contador de botellas
desechadas.

— OPERATE —
Contador_botellas2:=Contador_botellas2...

Etiquetas truncadas:

Etiqueta	Posición(es)
Contador_botellas2:=Contador_botellas2+1;	(8, 3)

Aumenta_cont_op <Acción> : [MAST - Embotelladora]



Aumenta_paquetes <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3											

Aumentamos en una unidad el número de paquetes realizados.

OPERATE

Contador_Paquetes:=Contador_Paquete...

Etiquetas truncadas:

Etiqueta	Posición(es)
Contador_Paquetes:=Contador_Paquetes+1;	(8, 3)

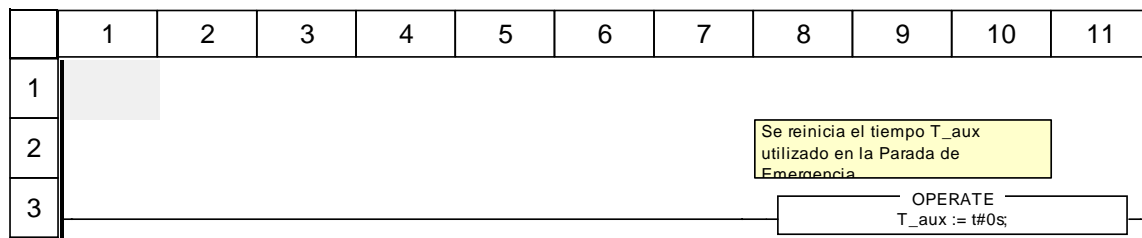
Vaciado_BotellasDesechadas <Acción> : [MAST - Embotelladora]

	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3											

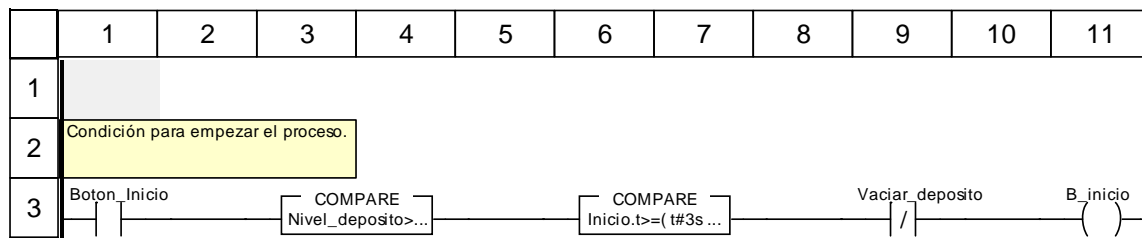
Se reinicia el contador de las botellas desechadas.

OPERATE
Contador_botellas2:=0;

Reiniciar_T_aux <Acción> : [MAST - Embotelladora]



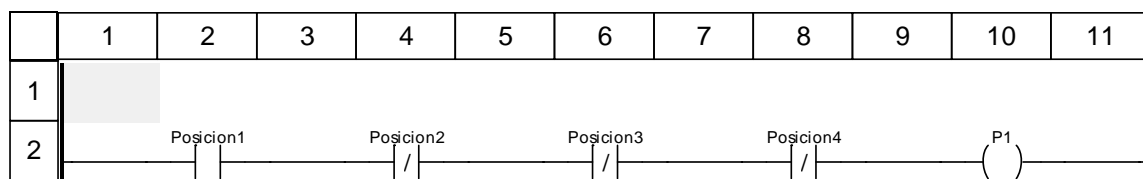
B_inicio <Transición> : [MAST - Embotelladora]



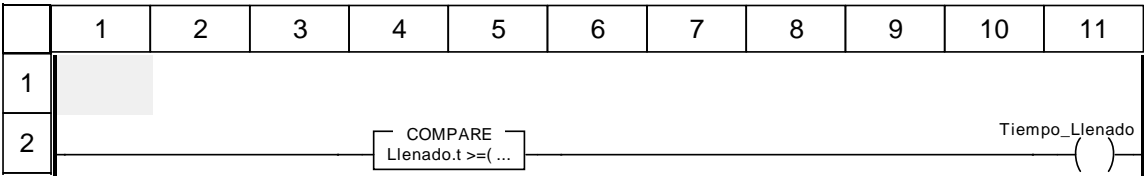
Etiquetas truncadas:

Etiqueta	Posición(es)
Inicio.t>=(t#3s + T_aux)	(6, 3)
Nivel_deposito>=20	(3, 3)

P1 <Transición> : [MAST - Embotelladora]



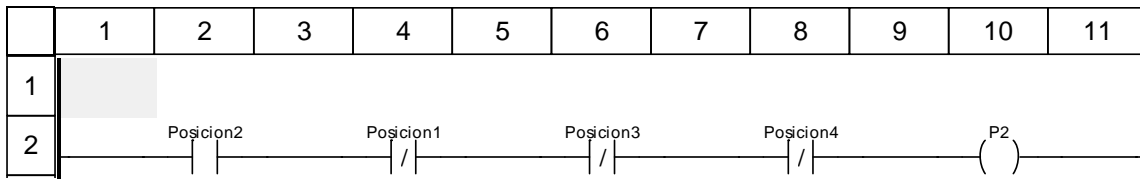
Tiempo_Llenado <Transición> : [MAST - Embotelladora]



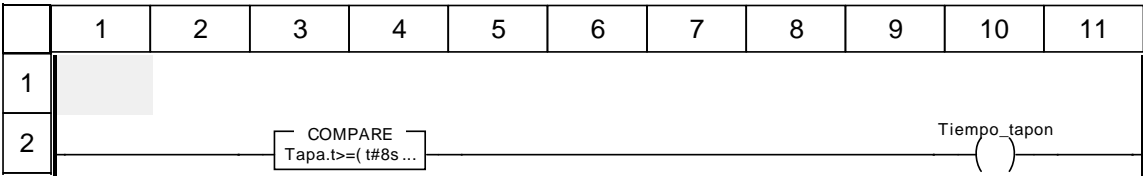
Etiquetas truncadas:

Etiqueta	Posición(es)
Llenado.t >= (t#5s+(T_aux))	(4, 2)

P2 <Transición> : [MAST - Embotelladora]



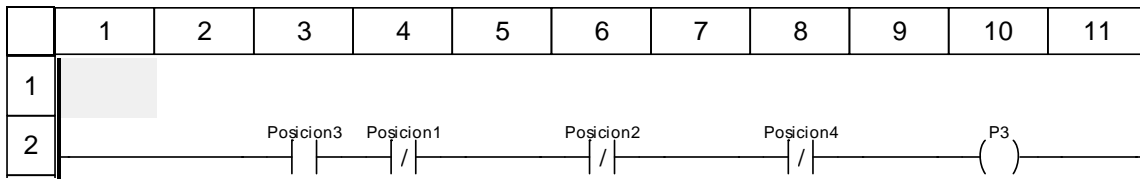
Tiempo_tapon <Transición> : [MAST - Embotelladora]



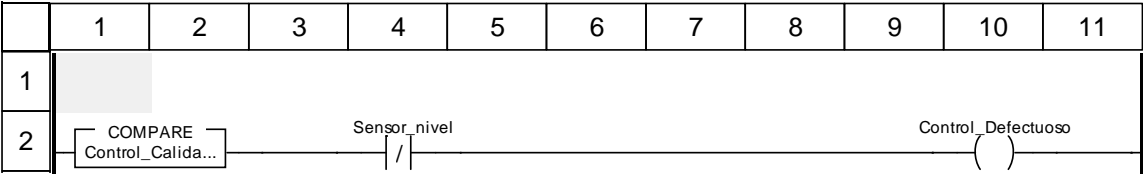
Etiquetas truncadas:

Etiqueta	Posición(es)
Tapa.t>=(t#8s + T_aux)	(3, 2)

P3 <Transición> : [MAST - Embotelladora]



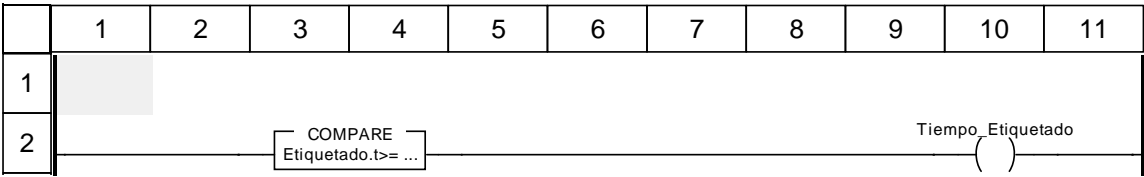
Control_Defectuoso <Transición> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Control_Calidad.t >= (t#8s + T_aux)	(1, 2)

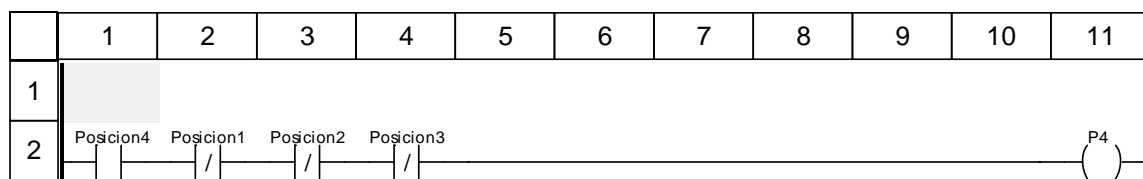
Tiempo_Etiquetado <Transición> : [MAST - Embotelladora]



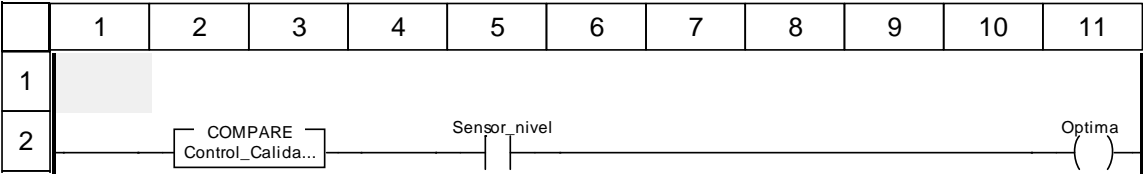
Etiquetas truncadas:

Etiqueta	Posición(es)
Etiquetado.t>= (t#18s + T_aux)	(3, 2)

P4 <Transición> : [MAST - Embotelladora]



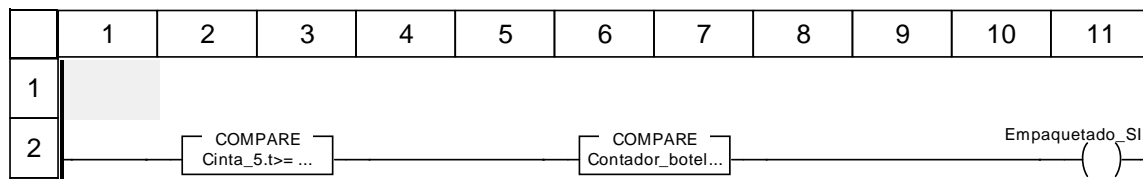
Optima <Transición> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Control_Calidad.t >= (t#8s + T_aux)	(2, 2)

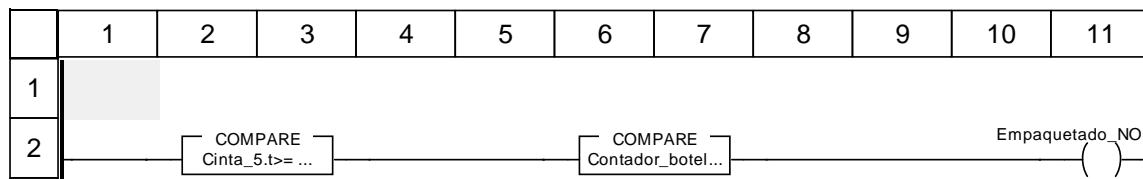
Empaquetado_SI <Transición> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Cinta_5.t>= (t#10s + T_aux)	(2, 2)
Contador_botellas = 3	(6, 2)

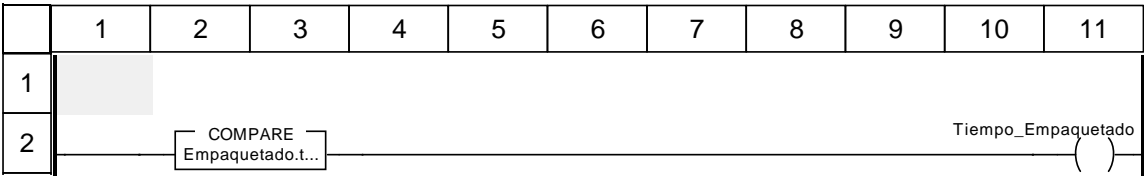
Empaquetado_NO <Transición> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Cinta_5.t>= (t#10s + T_aux)	(2, 2)
Contador_botellas <= 2	(6, 2)

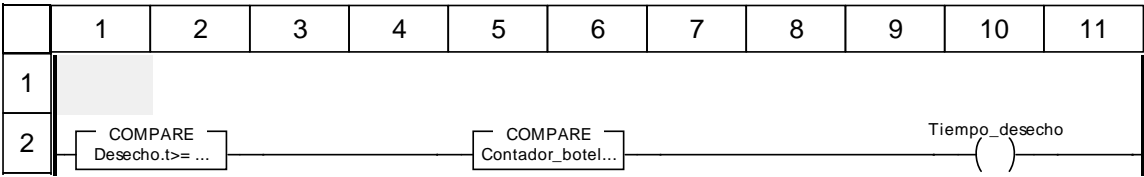
Tiempo_Empaquetado <Transición> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Empaquetado.t>= (t#5s + T_aux)	(2, 2)

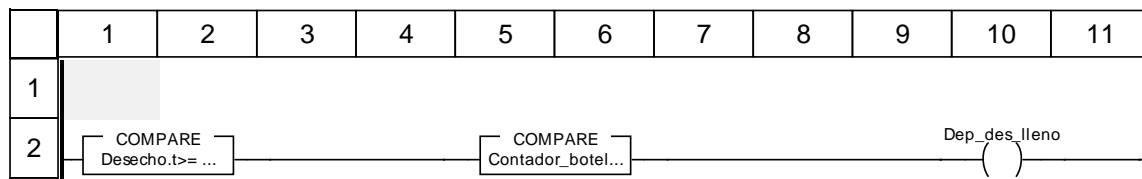
Tiempo_desecho <Transición> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Contador_botellas2<=2	(5, 2)
Desecho.t>= (t#10s + T_aux)	(1, 2)

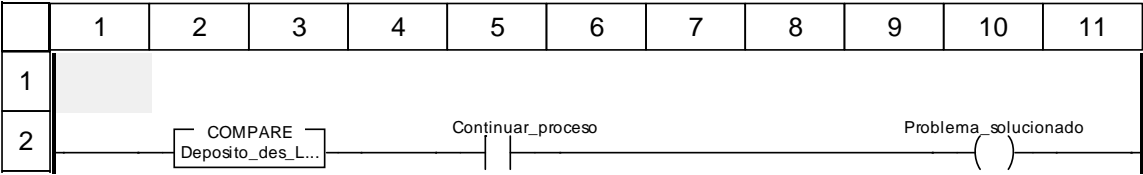
Dep_des_lleno <Transición> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Contador_botellas2<=3	(5, 2)
Desecho.t>= (t#10s + T_aux)	(1, 2)

Problema_solucionado <Transición> : [MAST - Embotelladora]



Etiquetas truncadas:

Etiqueta	Posición(es)
Deposito_des_Lleno.t>= (t#10s + T_aux)	(2, 2)

Control_deposito : [MAST]

Comentario

Propiedades comunes

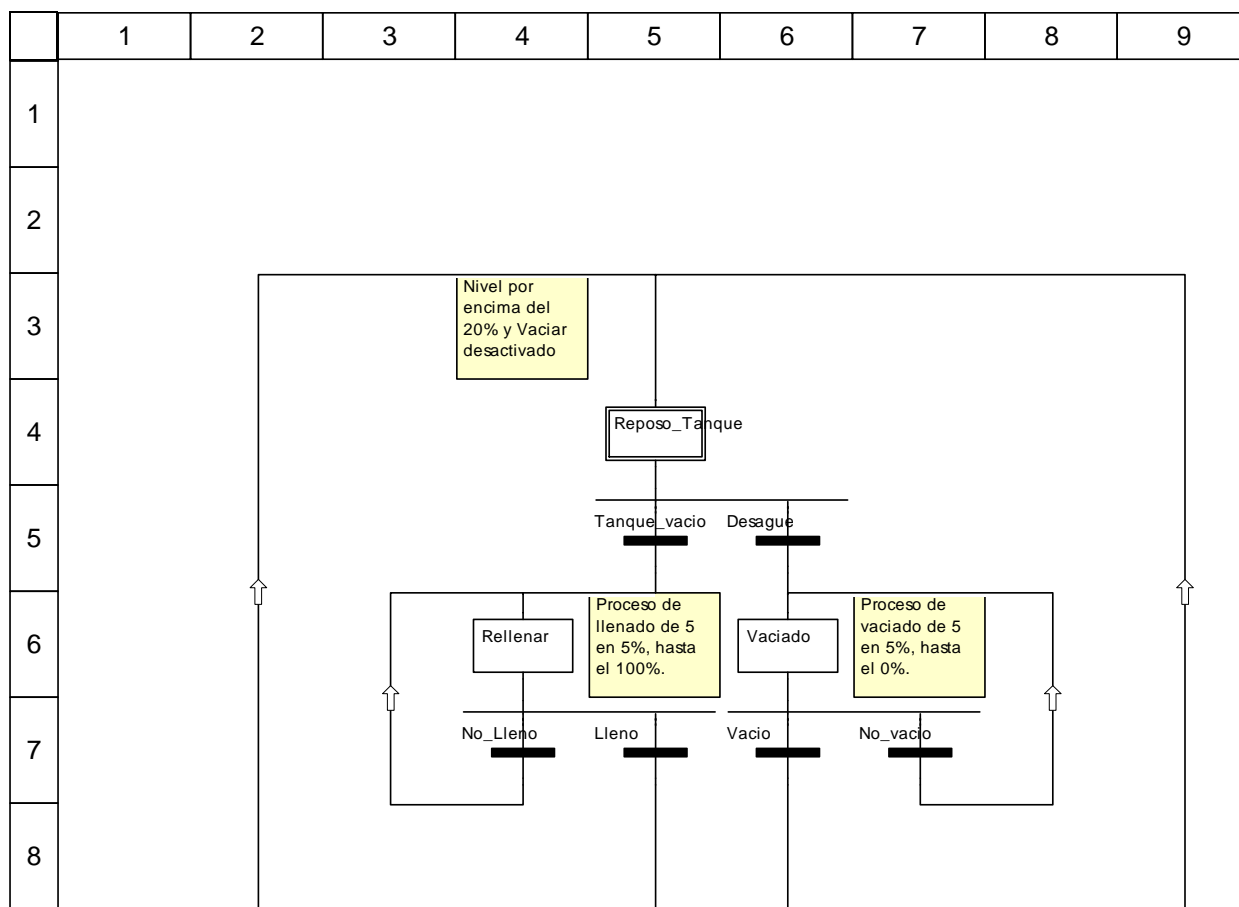
Módulo funcional	
Nombre de la condición	NOT Parada emergencia

Propiedades específicas

Control de operador	No
Número de área	0

Autor:	6.1.1.1 Secciones 6.1.1.1.2 Control_deposito	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 6.1.1.1.2 - 1/11

Chart : [MAST - Control_deposito]



Descripción de objeto

Pasos:

Rellenar			(4, 6)
Tiempo de supervisión mín./máx.:			Tiempo de retardo:
Comentario:			
Acciones:			
Descriptor: P1	Tiempo:	Sección: LD :: Llenar_tanque	
Descriptor: N	Tiempo:	Variable: V0	

Reposo_Tanque (paso inicial)			(5, 4)
Tiempo de supervisión mín./máx.:			Tiempo de retardo:
Comentario:			

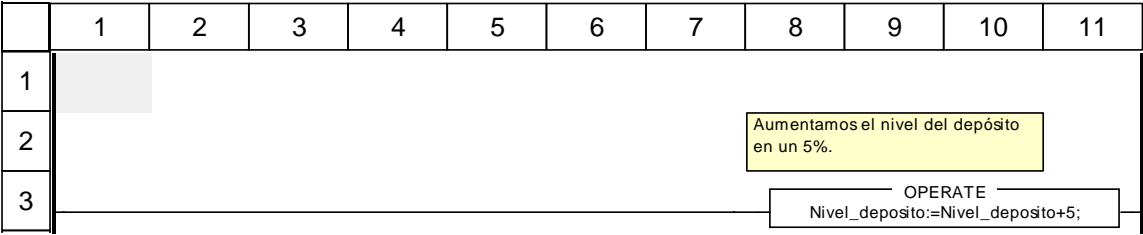
Vaciado			(6, 6)
Tiempo de supervisión mín./máx.:			Tiempo de retardo:
Comentario:			
Acciones:			
Descriptor: P1	Tiempo:	Sección: LD :: Vaciar_tanque	
Descriptor: N	Tiempo:	Variable: V desagüe	

Autor:	6.1.1.1.2 Control_deposito 6.1.1.1.2.1 Chart	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 6.1.1.1.2.1 - 1/2

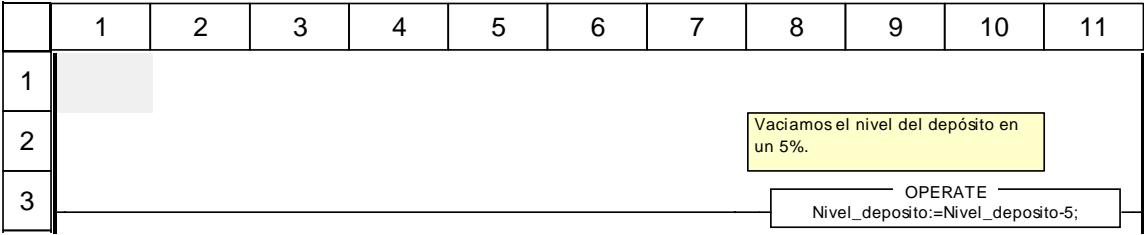
Transiciones:

Nombre	Tipo de condición	Posición	Comentario
LD :: Desague	Sección	(6, 5)	
LD :: Lleno	Sección	(5, 7)	
LD :: No Lleno	Sección	(4, 7)	
LD :: No vacío	Sección	(7, 7)	
LD :: Tanque_vacio	Sección	(5, 5)	
LD :: Vacío	Sección	(6, 7)	

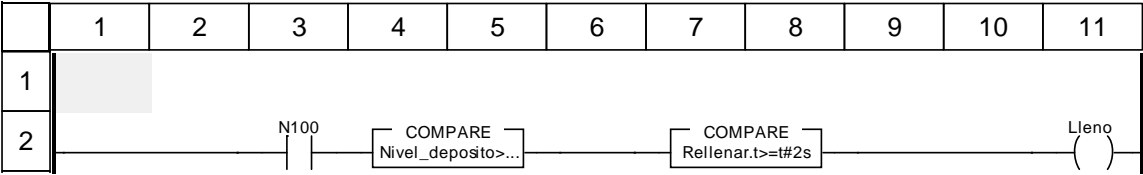
Llenar_tanque <Acción> : [MAST - Control_deposito]



Vaciar_tanque <Acción> : [MAST - Control_deposito]



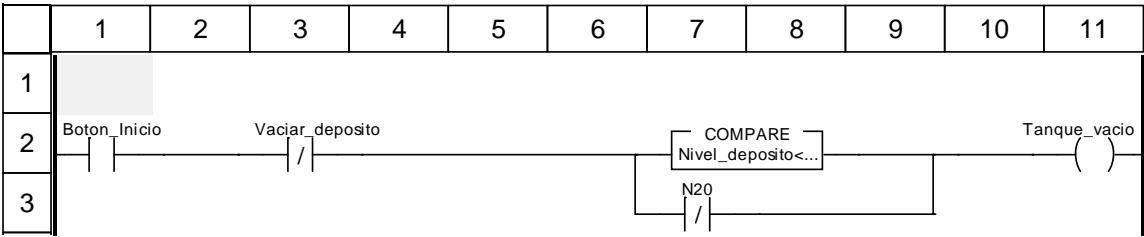
Lleno <Transición> : [MAST - Control_deposito]



Etiquetas truncadas:

Etiqueta	Posición(es)
Nivel_deposito>95	(4, 2)

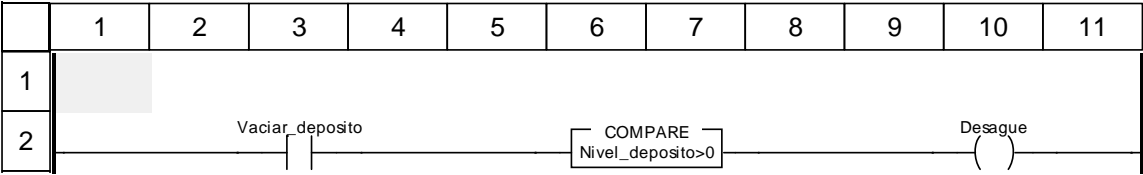
Tanque_vacio <Transición> : [MAST - Control_deposito]



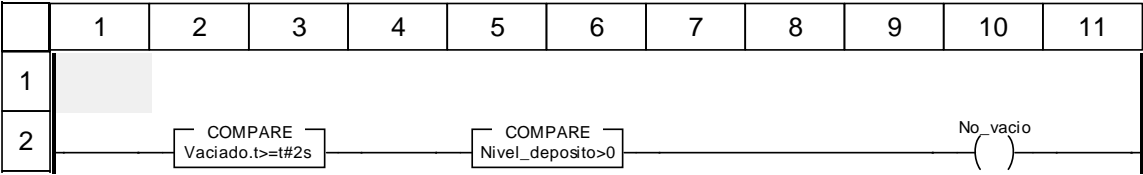
Etiquetas truncadas:

Etiqueta	Posición(es)
Nivel deposito<20	(7, 2)

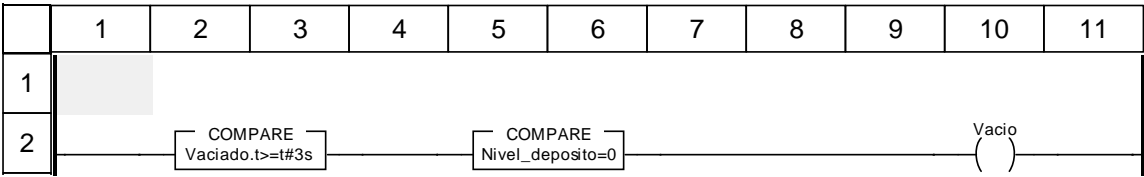
Desague <Transición> : [MAST - Control_deposito]



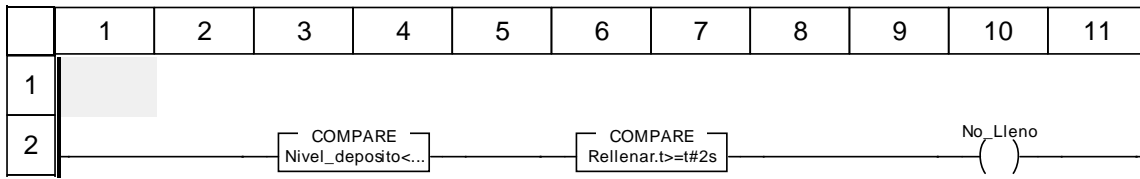
No_vacio <Transición> : [MAST - Control_deposito]



Vacio <Transición> : [MAST - Control_deposito]



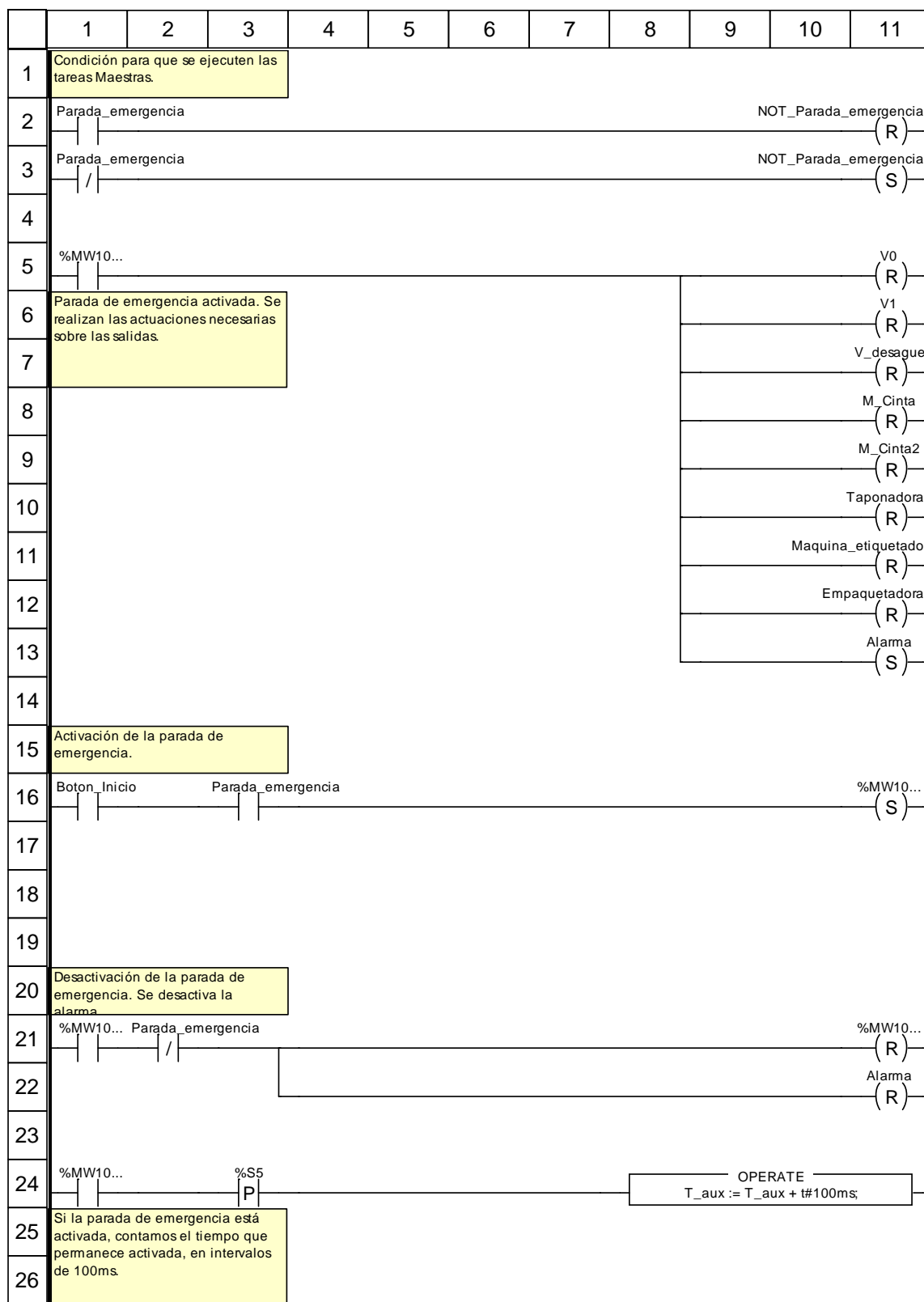
No_Lleno <Transición> : [MAST - Control_deposito]



Etiquetas truncadas:

Etiqueta	Posición(es)
Nivel_deposito<100	(3, 2)

Parada_de_Emergencia : [MAST]



Etiquetas truncadas:

Etiqueta	Posición(es)
----------	--------------

Autor:	6.1.1.1 Secciones	Impreso el 24/06/2013
Dept.:	6.1.1.1.3 Parada_de_Emergencia	
Proyecto: Sistema de Embotellado		Página: 6.1.1.1.3 - 1/2

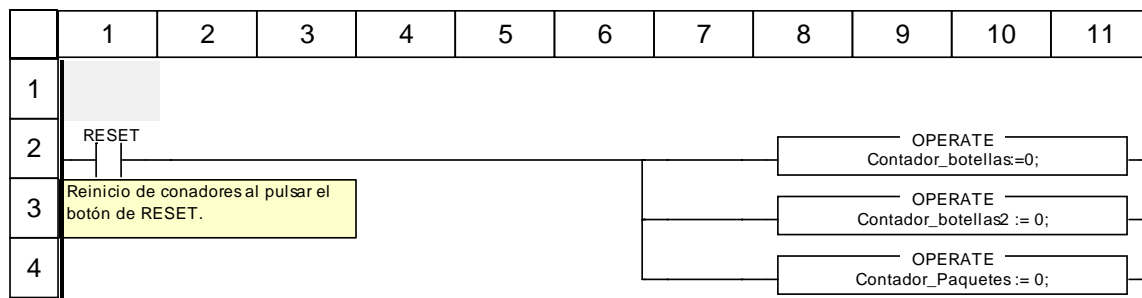
Este documento es propiedad de XXX y no se puede reproducir ni comercializar sin autorización previa.

%MW100.0	(1, 5) (11, 16) (1, 21) (11, 21) (1, 24)
----------	---

Autor:	6.1.1.1 Secciones 6.1.1.1.3 Parada_de_Emergencia	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 6.1.1.1.3 - 2/2

Este documento es propiedad de XXX y no se puede reproducir ni comercializar sin autorización previa.

Reset_contadores : [MAST]



Tablas de animación

Nombre de tabla: Entradas

Comentario de tabla:

Módulo funcional:

Nombre	Tipo	Comentario
Boton Inicio	EBOOL	Boton para iniciar el proceso.
Posicion1	EBOOL	Sensor que indica que la botella ha llegado a la zona de llenado.
Posicion2	EBOOL	Sensor que indica que la botella esta en la zona de taponado.
Posicion3	EBOOL	Sensor que indica que la botella esta en la posición de etiquetado.
Sensor_nivel	EBOOL	Sensor que indica si la botella contiene el líquido mínimo para pasar el control de calidad.
N20	EBOOL	Sensor de nivel que indica que el tanque está por encima del 20%
N100	EBOOL	Sensor de nivel que indica que el tanque esta lleno.
RESET	EBOOL	Boton Reset.
Parada emergencia	EBOOL	Pulsador de Pardada de emergencia.
NOT_Parada_emergencia	BOOL	NOT(Parada_Emergencia). Condición para que se produzcan las transiciones en los SFC.
T_aux2	INT	Almacena como entero el tiempo de T_aux.
T_aux	TIME	Guarda el tiempoque está activa la parada de emergencia.
Posicion4	EBOOL	Sensor que indica que la botella ha llegado a la zona de control de calidad.
Continuar_proceso	EBOOL	Botón que se pulsará para indicar que se ha subsanado cualquier error.
Vaciar_deposito	EBOOL	Botón utilizado para vaciar el contenido del tanque.

Tablas de animación

Nombre de tabla: Salidas

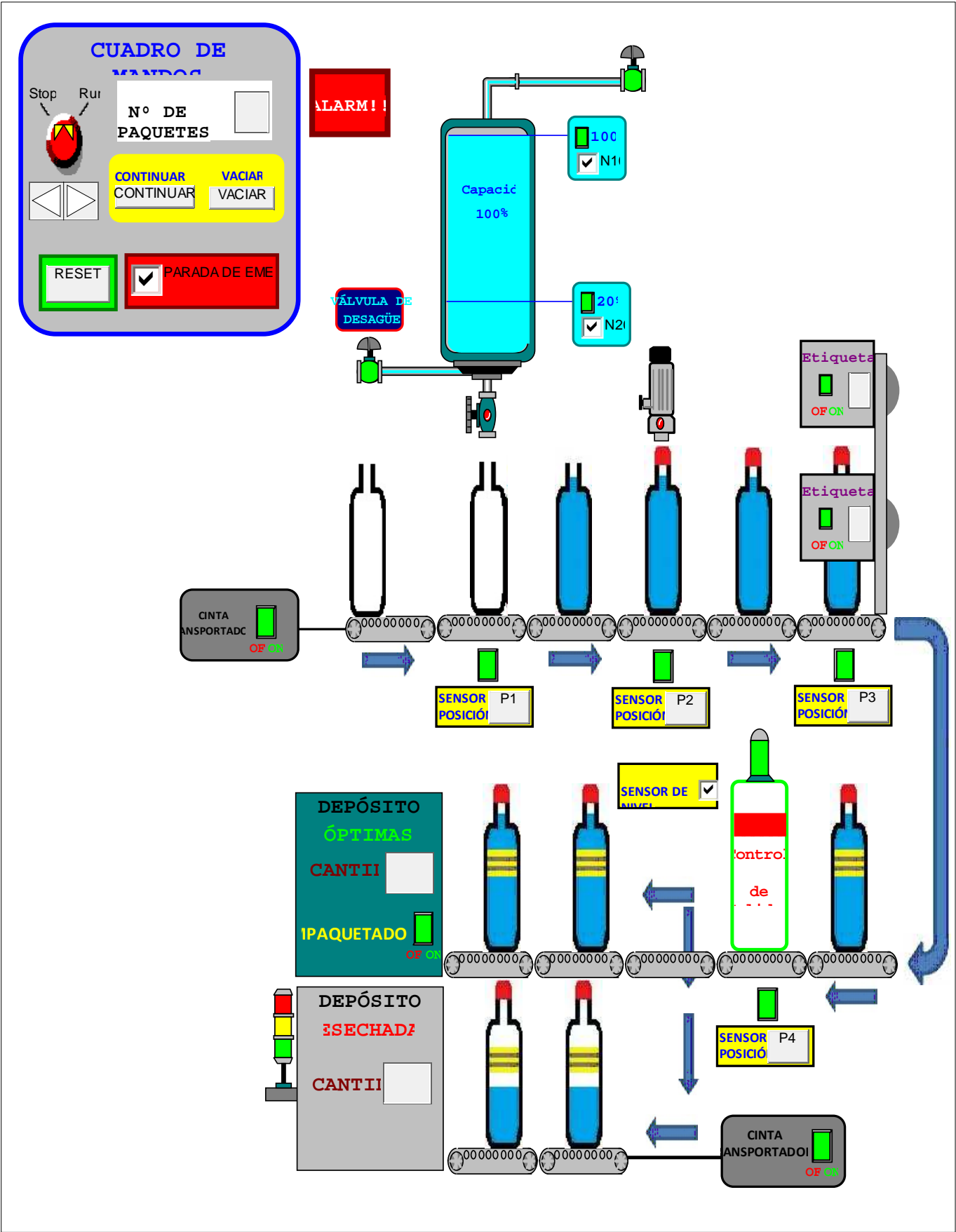
Comentario de tabla:

Módulo funcional:

Nombre	Tipo	Comentario
M_Cinta2	EBOOL	Cinta transportadora que deshecha las botellas malas,
M_Cinta	EBOOL	Motor que mueve la cinta transportadora principal.
V1	EBOOL	Válvula que abre el tanque que contiene el fluido.
Taponadora	EBOOL	Proceso de colocar el tapón en la botella.
V0	EBOOL	Válvula que llena el tanque que contiene el fluido.
V_desague	EBOOL	Válvula que destina el contenido del tanque hacia el desagüe.
Maquina_etiquetado	EBOOL	Máquina para el etiquetado de las botellas.
Posicion_etiquetadora	EBOOL	Motor para mover la posición de la máquina de etiquetado. 1 Abajo, 0 Arriba.
Alarma	EBOOL	Señal acústica que se activa en caso de que se detecte algún problema.
Empaquetadora	EBOOL	Máquina encargada del empaquetado de botellas.

Autor:	7 Tablas de animación 7.2 Salidas	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 7.2 - 1/1

Instalacion



Eje de movimiento

Autor:	9 Movimiento	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 9 - 1/1

Este documento es propiedad de XXX y no se puede reproducir ni comercializar sin autorización previa.

Referencias cruzadas

Aplicación:

Direcciones

Objeto	Referenciado dentro	Etiqueta	Uso
%MW100	(No utilizado directamente)		
(%MW100.0)	Parada de Emergencia : [MAST]	(l: 21, c: 1)	L
		(l: 24, c: 1)	L
		(l: 21, c: 11)	E
		(l: 5, c: 1)	L
		(l: 16, c: 11)	E
%MW20	(No utilizado directamente)		
(%MW20.0)	Instalacion		L (x104)
%S5	Parada de Emergencia : [MAST]	(l: 24, c: 3)	L

Variables o instancias FB

Objeto	Referenciado dentro	Etiqueta	Uso
Alarma	<Pantalla>Instalacion		L (x1)
	Chart : [MAST - Embotelladora]	(l: 23, c: 3)	E
	Parada de Emergencia : [MAST]	(l: 22, c: 11)	E
		(l: 13, c: 11)	E
B_inicio	Chart : [MAST - Embotelladora]	(l: 4, c: 3)	L
	B_inicio <Transición> : [MAST - Embotelladora]	(l: 3, c: 11)	E
Boton_Inicio	<Pantalla>Instalacion		L (x2)
			L/E (x1)
	B_inicio <Transición> : [MAST - Embotelladora]	(l: 3, c: 1)	L
	Parada de Emergencia : [MAST]	(l: 16, c: 1)	L
	Tanque_vacio <Transición> : [MAST - Control deposito]	(l: 2, c: 1)	L
Cinta_1	Chart : [MAST - Embotelladora]	(l: 5, c: 3)	E
Cinta_2	Chart : [MAST - Embotelladora]	(l: 9, c: 3)	E
	Botella4 <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
Cinta_3	Chart : [MAST - Embotelladora]	(l: 13, c: 3)	E
	Botella7 <Acción> : [MAST - Embotelladora]	(l: 3, c: 2)	L
Cinta_4	Chart : [MAST - Embotelladora]	(l: 17, c: 3)	E
	Botella10 <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
Cinta_5	Empaquetado_SI <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
	Botella12 <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
	Empaquetado_NO <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
	Chart : [MAST - Embotelladora]	(l: 21, c: 4)	E
	Botella13 <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
Contador_Paquetes	<Pantalla>Instalacion		L/E (x1)
	Reset_contadores : [MAST]	(l: 4, c: 8)	E
	Aumenta_paquetes <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	L
		(l: 3, c: 8)	E
Contador_botellas	<Pantalla>Instalacion		L/E (x1)
	Empaquetado_SI <Transición> : [MAST - Embotelladora]	(l: 2, c: 6)	L
	Empaquetado_NO <Transición> : [MAST - Embotelladora]	(l: 2, c: 6)	L
	Reset_contadores : [MAST]	(l: 2, c: 8)	E
	Aumenta_cont_op <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	L
		(l: 3, c: 8)	E

Autor:	10 Referencias cruzadas	Impreso el 24/06/2013
Dept.:		
Proyecto: Sistema de Embotellado		Página: 10 - 1/7

Referencias cruzadas

Objeto	Referenciado dentro	Etiqueta	Uso
	Reiniciar_contador <Acción> : [MAST - Embotelladora]	(l: 2, c: 8)	E
Contador_botellas2	<Pantalla>Instalacion		L (x3)
			L/E (x1)
	Tiempo_desecho <Transición> : [MAST - Embotelladora]	(l: 2, c: 5)	L
	Reset_contadores : [MAST]	(l: 3, c: 8)	E
	Dep_des_lleno <Transición> : [MAST - Embotelladora]	(l: 2, c: 5)	L
	Vaciado_BotellasDesechadas <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Aumeta_cont_des <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	L
		(l: 3, c: 8)	E
Continuar_proceso	<Pantalla>Instalacion		L/E (x1)
	Problema_solucionado <Transición> : [MAST - Embotelladora]	(l: 2, c: 5)	L
Control_Defectuoso	Chart : [MAST - Embotelladora]	(l: 20, c: 3)	L
	Control_Defectuoso <Transición> : [MAST - Embotelladora]	(l: 2, c: 10)	E
Control_calidad	<Pantalla>Instalacion		L/E (x1)
	Chart : [MAST - Embotelladora]	(l: 19, c: 3)	E
	Optima <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
	Control_Defectuoso <Transición> : [MAST - Embotelladora]	(l: 2, c: 1)	L
Dep_des_lleno	Chart : [MAST - Embotelladora]	(l: 22, c: 3)	L
	Dep_des_lleno <Transición> : [MAST - Embotelladora]	(l: 2, c: 10)	E
Deposito_des_Lleno	Chart : [MAST - Embotelladora]	(l: 23, c: 3)	E
	Problema_solucionado <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
Desague	Desague <Transición> : [MAST - Control deposito]	(l: 2, c: 10)	E
	Chart : [MAST - Control deposito]	(l: 5, c: 6)	L
Desecho	Botella14 <Acción> : [MAST - Embotelladora]	(l: 3, c: 2)	L
	Tiempo_desecho <Transición> : [MAST - Embotelladora]	(l: 2, c: 1)	L
	Dep_des_lleno <Transición> : [MAST - Embotelladora]	(l: 2, c: 1)	L
	Chart : [MAST - Embotelladora]	(l: 21, c: 2)	E
	Botella15 <Acción> : [MAST - Embotelladora]	(l: 3, c: 2)	L
Empaquetado	Tiempo_Empaquetado <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
	Chart : [MAST - Embotelladora]	(l: 23, c: 4)	E
Empaquetado_NO	Chart : [MAST - Embotelladora]	(l: 22, c: 5)	L
	Empaquetado_NO <Transición> : [MAST - Embotelladora]	(l: 2, c: 11)	E
Empaquetado_SI	Chart : [MAST - Embotelladora]	(l: 22, c: 4)	L
	Empaquetado_SI <Transición> : [MAST - Embotelladora]	(l: 2, c: 11)	E
Empaquetadora	<Pantalla>Instalacion		L (x4)
	Chart : [MAST - Embotelladora]	(l: 23, c: 4)	E
	Parada_de_Emergencia : [MAST]	(l: 12, c: 11)	E
Etiquetado	Chart : [MAST - Embotelladora]	(l: 15, c: 3)	E
	Maq_etiquetadora <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
		(l: 4, c: 2)	L
	Botella9 <Acción> : [MAST - Embotelladora]	(l: 3, c: 1)	L
		(l: 3, c: 1)	L

Referencias cruzadas

Objeto	Referenciado dentro	Etiqueta	Uso
	Etiquetado_Time <Acción> : [MAST - Embotelladora]	(l: 2, c: 5)	L
	Proceso_etiquetado <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
		(l: 4, c: 2)	L
	Tiempo_Etiquetado <Transición> : [MAST - Embotelladora]	(l: 2, c: 3)	L
Inicio	B_inicio <Transición> : [MAST - Embotelladora]	(l: 3, c: 6)	L
	Chart : [MAST - Embotelladora]	(l: 3, c: 3)	E
Llenado	Chart : [MAST - Embotelladora]	(l: 7, c: 3)	E
	Tiempo_Llenado <Transición> : [MAST - Embotelladora]	(l: 2, c: 4)	L
	Botella2 <Acción> : [MAST - Embotelladora]	(l: 3, c: 2)	L
Lleno	Lleno <Transición> : [MAST - Control deposito]	(l: 2, c: 11)	E
	Chart : [MAST - Control deposito]	(l: 7, c: 5)	L
M_Cinta	<Pantalla>Instalacion		L (x10)
	Chart : [MAST - Embotelladora]	(l: 5, c: 3)	E
		(l: 9, c: 3)	E
		(l: 13, c: 3)	E
		(l: 17, c: 3)	E
		(l: 21, c: 4)	E
	Parada de Emergencia : [MAST]	(l: 8, c: 11)	E
M_Cinta2	<Pantalla>Instalacion		L (x7)
	Parada de Emergencia : [MAST]	(l: 9, c: 11)	E
	Chart : [MAST - Embotelladora]	(l: 21, c: 2)	E
Maquina_etiquetado	<Pantalla>Instalacion		L (x8)
	Chart : [MAST - Embotelladora]	(l: 15, c: 3)	E
	Maq_etiquetadora <Acción> : [MAST - Embotelladora]	(l: 4, c: 10)	E
	Parada de Emergencia : [MAST]	(l: 11, c: 11)	E
N100	<Pantalla>Instalacion		L/E (x1)
	Lleno <Transición> : [MAST - Control deposito]	(l: 2, c: 3)	L
N20	<Pantalla>Instalacion		L/E (x1)
	Tanque_vacio <Transición> : [MAST - Control deposito]	(l: 3, c: 7)	L
NOT_Parada_emergencia	Embotelladora : [MAST]	Condición de activación: NOT_Parada_emergencia	L
	Control_deposito : [MAST]	Condición de activación: NOT_Parada_emergencia	L
	Parada de Emergencia : [MAST]	(l: 3, c: 11)	E
		(l: 2, c: 11)	E
Nivel_deposito	Vaciar_tanque <Acción> : [MAST - Control deposito]	(l: 3, c: 8)	L
		(l: 3, c: 8)	E
	Llenar_tanque <Acción> : [MAST - Control deposito]	(l: 3, c: 8)	L
		(l: 3, c: 8)	E
	No_Lleno <Transición> : [MAST - Control deposito]	(l: 2, c: 3)	L
	<Pantalla>Instalacion		L (x6)
	B_inicio <Transición> : [MAST - Embotelladora]	(l: 3, c: 3)	L
	Decrementar_Deposito <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	L
		(l: 3, c: 8)	E

Referencias cruzadas

Objeto	Referenciado dentro	Etiqueta	Uso
	Vacio <Transición> : [MAST - Control deposito]	(l: 2, c: 5)	L
	No_vacio <Transición> : [MAST - Control deposito]	(l: 2, c: 5)	L
	Desague <Transición> : [MAST - Control deposito]	(l: 2, c: 6)	L
	Tanque_vacio <Transición> : [MAST - Control deposito]	(l: 2, c: 7)	L
	Lleno <Transición> : [MAST - Control deposito]	(l: 2, c: 4)	L
No_Lleno	Chart : [MAST - Control deposito]	(l: 7, c: 4)	L
	No_Lleno <Transición> : [MAST - Control deposito]	(l: 2, c: 10)	E
No_vacio	No_vacio <Transición> : [MAST - Control deposito]	(l: 2, c: 10)	E
	Chart : [MAST - Control deposito]	(l: 7, c: 7)	L
Optima	Chart : [MAST - Embotelladora]	(l: 20, c: 4)	L
	Optima <Transición> : [MAST - Embotelladora]	(l: 2, c: 11)	E
P1	Chart : [MAST - Embotelladora]	(l: 6, c: 3)	L
	P1 <Transición> : [MAST - Embotelladora]	(l: 2, c: 10)	E
P2	Chart : [MAST - Embotelladora]	(l: 10, c: 3)	L
	P2 <Transición> : [MAST - Embotelladora]	(l: 2, c: 10)	E
P3	P3 <Transición> : [MAST - Embotelladora]	(l: 2, c: 10)	E
	Chart : [MAST - Embotelladora]	(l: 14, c: 3)	L
P4	Chart : [MAST - Embotelladora]	(l: 18, c: 3)	L
	P4 <Transición> : [MAST - Embotelladora]	(l: 2, c: 11)	E
Parada_emergencia	<Pantalla>Instalacion		L/E (x1)
	Parada de Emergencia : [MAST]	(l: 2, c: 1)	L
		(l: 3, c: 1)	L
		(l: 21, c: 2)	L
		(l: 16, c: 3)	L
Posicion	Sin_botella <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella14 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella12 <Acción> : [MAST - Embotelladora]	(l: 4, c: 8)	E
	Botella7 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella13 <Acción> : [MAST - Embotelladora]	(l: 4, c: 8)	E
	<Pantalla>Instalacion		L (x15)
	Botella11 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella8 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella9 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella6 <Acción> : [MAST - Embotelladora]	(l: 4, c: 8)	E
	Botella2 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella4 <Acción> : [MAST - Embotelladora]	(l: 4, c: 8)	E
	Botella10 <Acción> : [MAST - Embotelladora]	(l: 4, c: 8)	E
	Botella1 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella0 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella5 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella15 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella16 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella3 <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
Posicion1	<Pantalla>Instalacion		L (x2)
			L/E (x1)
	P1 <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
	P4 <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
	P3 <Transición> : [MAST - Embotelladora]	(l: 2, c: 4)	L
	Botella4 <Acción> : [MAST - Embotelladora]	(l: 4, c: 5)	L
	P2 <Transición> : [MAST - Embotelladora]	(l: 2, c: 4)	L
Posicion2	<Pantalla>Instalacion		L (x2)
			L/E (x1)
	P1 <Transición> : [MAST - Embotelladora]	(l: 2, c: 4)	L

Referencias cruzadas

Objeto	Referenciado dentro	Etiqueta	Uso
	Botella7 <Acción> : [MAST - Embotelladora]	(l: 3, c: 5)	L
	P4 <Transición> : [MAST - Embotelladora]	(l: 2, c: 3)	L
	P3 <Transición> : [MAST - Embotelladora]	(l: 2, c: 6)	L
	P2 <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
Posicion3	<Pantalla>Instalacion		L (x2)
			L/E (x1)
	P1 <Transición> : [MAST - Embotelladora]	(l: 2, c: 6)	L
	P4 <Transición> : [MAST - Embotelladora]	(l: 2, c: 4)	L
	P3 <Transición> : [MAST - Embotelladora]	(l: 2, c: 3)	L
	Botella10 <Acción> : [MAST - Embotelladora]	(l: 4, c: 5)	L
	P2 <Transición> : [MAST - Embotelladora]	(l: 2, c: 6)	L
Posicion4	P2 <Transición> : [MAST - Embotelladora]	(l: 2, c: 8)	L
	Botella13 <Acción> : [MAST - Embotelladora]	(l: 4, c: 5)	L
	P1 <Transición> : [MAST - Embotelladora]	(l: 2, c: 8)	L
	<Pantalla>Instalacion		L (x2)
			L/E (x1)
	Botella12 <Acción> : [MAST - Embotelladora]	(l: 4, c: 4)	L
	P4 <Transición> : [MAST - Embotelladora]	(l: 2, c: 1)	L
	P3 <Transición> : [MAST - Embotelladora]	(l: 2, c: 8)	L
Posicion etiquetadora	<Pantalla>Instalacion		L (x2)
	Chart : [MAST - Embotelladora]	(l: 15, c: 3)	E
	Proceso_etiquetado <Acción> : [MAST - Embotelladora]	(l: 4, c: 11)	E
Problema solucionado	Chart : [MAST - Embotelladora]	(l: 24, c: 3)	L
	Problema_solucionado <Transición> : [MAST - Embotelladora]	(l: 2, c: 10)	E
RESET	<Pantalla>Instalacion		L/E (x1)
	Reset contadores : [MAST]	(l: 2, c: 1)	L
Rellenar	No Lleno <Transición> : [MAST - Control deposito]	(l: 2, c: 6)	L
	Chart : [MAST - Control deposito]	(l: 6, c: 4)	E
	Lleno <Transición> : [MAST - Control deposito]	(l: 2, c: 7)	L
Reposo Tanque	Chart : [MAST - Control deposito]	(l: 4, c: 5)	E
Sensor nivel	<Pantalla>Instalacion		L (x3)
			L/E (x1)
	Control_Defectuoso <Transición> : [MAST - Embotelladora]	(l: 2, c: 4)	L
	Optima <Transición> : [MAST - Embotelladora]	(l: 2, c: 5)	L
T_aux	Botella14 <Acción> : [MAST - Embotelladora]	(l: 3, c: 2)	L
	Problema_solucionado <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
	Botella12 <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
	Dep_des_lleno <Transición> : [MAST - Embotelladora]	(l: 2, c: 1)	L
	Botella7 <Acción> : [MAST - Embotelladora]	(l: 3, c: 2)	L
	Empaquetado_SI <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
	Etiquetado_Time <Acción> : [MAST - Embotelladora]	(l: 6, c: 5)	L
	Tiempo_desecho <Transición> : [MAST - Embotelladora]	(l: 2, c: 1)	L
	Botella10 <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
	Optima <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
	Botella13 <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
	Botella9 <Acción> : [MAST - Embotelladora]	(l: 3, c: 1)	L
		(l: 3, c: 1)	L
	Empaquetado_NO <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L

Referencias cruzadas

Objeto	Referenciado dentro	Etiqueta	Uso
	Tiempo_tapon <Transición> : [MAST - Embotelladora]	(l: 2, c: 3)	L
	Botella4 <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
	Tiempo_Llenado <Transición> : [MAST - Embotelladora]	(l: 2, c: 4)	L
	Parada de Emergencia : [MAST]	(l: 24, c: 8)	L
		(l: 24, c: 8)	E
	Botella2 <Acción> : [MAST - Embotelladora]	(l: 3, c: 2)	L
	B_inicio <Transición> : [MAST - Embotelladora]	(l: 3, c: 6)	L
	Control Defectuoso <Transición> : [MAST - Embotelladora]	(l: 2, c: 1)	L
	Proceso_etiquetado <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
		(l: 4, c: 2)	L
	Tiempo_Etiquetado <Transición> : [MAST - Embotelladora]	(l: 2, c: 3)	L
	Reiniciar_T_aux <Acción> : [MAST - Embotelladora]	(l: 3, c: 8)	E
	Botella15 <Acción> : [MAST - Embotelladora]	(l: 3, c: 2)	L
	Maq_etiquetadora <Acción> : [MAST - Embotelladora]	(l: 4, c: 2)	L
		(l: 4, c: 2)	L
	Tiempo_Empaquetado <Transición> : [MAST - Embotelladora]	(l: 2, c: 2)	L
T_aux2	Etiquetado_Time <Acción> : [MAST - Embotelladora]	(l: 10, c: 8)	L
		(l: 10, c: 1)	L
		(l: 10, c: 1)	L
		(l: 6, c: 5)	E
Tanque vacio	Chart : [MAST - Control deposito]	(l: 5, c: 5)	L
	Tanque_vacio <Transición> : [MAST - Control deposito]	(l: 2, c: 11)	E
Tapa	Chart : [MAST - Embotelladora]	(l: 11, c: 3)	E
	Tiempo_tapon <Transición> : [MAST - Embotelladora]	(l: 2, c: 3)	L
Taponadora	<Pantalla>Instalacion		L (x2)
	Chart : [MAST - Embotelladora]	(l: 11, c: 3)	E
	Parada de Emergencia : [MAST]	(l: 10, c: 11)	E
Tiempo_Empaquetado	Tiempo_Empaquetado <Transición> : [MAST - Embotelladora]	(l: 2, c: 11)	E
	Chart : [MAST - Embotelladora]	(l: 24, c: 4)	L
Tiempo_Etiquetado	Chart : [MAST - Embotelladora]	(l: 16, c: 3)	L
	Tiempo_Etiquetado <Transición> : [MAST - Embotelladora]	(l: 2, c: 10)	E
Tiempo_Llenado	Chart : [MAST - Embotelladora]	(l: 8, c: 3)	L
	Tiempo_Llenado <Transición> : [MAST - Embotelladora]	(l: 2, c: 11)	E
Tiempo_desecho	Chart : [MAST - Embotelladora]	(l: 22, c: 2)	L
	Tiempo_desecho <Transición> : [MAST - Embotelladora]	(l: 2, c: 10)	E
Tiempo_etiquetadora	<Pantalla>Instalacion		L/E (x2)
	Etiquetado_Time <Acción> : [MAST - Embotelladora]	(l: 10, c: 8)	E
Tiempo_etiquetadora2	Etiquetado_Time <Acción> : [MAST - Embotelladora]	(l: 2, c: 5)	E
		(l: 10, c: 8)	L
		(l: 10, c: 1)	L
		(l: 10, c: 1)	L
Tiempo_tapon	Chart : [MAST - Embotelladora]	(l: 12, c: 3)	L

Referencias cruzadas

Objeto	Referenciado dentro	Etiqueta	Uso
V0	Tiempo_tapon <Transición> : [MAST - Embotelladora]	(l: 2, c: 10)	E
	<Pantalla>Instalacion		L (x5)
	Chart : [MAST - Control deposito]	(l: 6, c: 4)	E
V1	Parada de Emergencia : [MAST]	(l: 5, c: 11)	E
	<Pantalla>Instalacion		L (x2)
	Chart : [MAST - Embotelladora]	(l: 7, c: 3)	E
V_desague	Parada de Emergencia : [MAST]	(l: 6, c: 11)	E
	<Pantalla>Instalacion		L (x3)
	Chart : [MAST - Control deposito]	(l: 6, c: 6)	E
Vaciado	Parada de Emergencia : [MAST]	(l: 7, c: 11)	E
	No_vacio <Transición> : [MAST - Control deposito]	(l: 2, c: 2)	L
	Chart : [MAST - Control deposito]	(l: 6, c: 6)	E
Vaciar_deposito	Vacio <Transición> : [MAST - Control deposito]	(l: 2, c: 2)	L
	<Pantalla>Instalacion		L/E (x1)
	Desague <Transición> : [MAST - Control deposito]	(l: 2, c: 3)	L
Vacio	B_inicio <Transición> : [MAST - Embotelladora]	(l: 3, c: 9)	L
	Tanque_vacio <Transición> : [MAST - Control deposito]	(l: 2, c: 3)	L
	Vacio <Transición> : [MAST - Control deposito]	(l: 2, c: 10)	E
	Chart : [MAST - Control deposito]	(l: 7, c: 6)	L

Objetos EF

Objeto	Referenciado dentro	Etiqueta	Uso
time_to_int	Etiquetado_Time <Acción> : [MAST - Embotelladora]	(l: 2, c: 5)	LLAM F
		(l: 6, c: 5)	LLAM F

Subrutinas

Objeto	Referenciado dentro	Etiqueta	Uso
--------	---------------------	----------	-----

Hipervínculos y directorios de usuario

